Working
Draft

T13
1510D

Revision 1.0
January 17, 2003

ATA/ATAPI Host Adapters Standard (ATA – Adapter)

This is an internal working document of T13, a Technical Committee of Accredited Standards Committee INCITS. The T13 Technical Committee may modify the contents. This document is made available for review and comment only.

T13 Technical Editor:

Tony Goodfellow
Pacific Digital Corporation
2052 Alton Parkway
Irvine, CA92602
USA
Tel: 949-252-1111
Fax: 949-252-9397
Email: tgoodfellow@pacificdigital.com

Other Points of Contact:

T13 Chair	T13 Vicechair
Pete McLean	Dan Colegrove
Maxtor Corporation	IBM Corporation
2452 Clover Basin Drive	2903 Carmelo Drive
Longmont, CO 80501	Henderson, NV 89052
Tel: 303-678-2149	Tel: 702 614-6119
Fax: 303-682-4811	Fax: 702 614-7995

INCITS Secretariat
Administrator Standards Processing
1250 Eye Street, NW Suite 200
Washington, DC 20005
Tel: 202-737-8888
Fax: 202-638-4922
Email: INCITS@ITIC.ORG

T13 Reflector
Internet address for subscription to the T13 reflector: majordomo@t13.org
Send email to above account and include in BODY of text, on a line by itself the following:
    "subscribe T13"
Internet address for unsubscribing to the T13 reflector: majordomo@t13.org
Send email to above account and include in BODY of text, on a line by itself the following:
    "unsubscribe T13"

Internet address for distribution via T13 reflector: T13@t13.org

T13 Web Site

http://www.t13.org

T13 Anonymous FTP Site

ftp.t13.org

T13 mailings

Global Engineering
15 Inverness Way East
Englewood, CO   80112-5704
Tel: 303-792-2181 or 800-854-7179
Fax: 303-792-2192

DOCUMENT STATUS

Revision 0 – 20 October 2000
Document created.
Revision 0a - 7 August 2001
Added new description of ADMA mode.
Revision 0b – 23 October 2001
Made editorial corrections and removed descriptive elements not compatible with a standard.
Revision 0c – 7 December 2001
Editorial changes from editorial review at October 2001 plenary meeting.
Revision 0d – February 25, 2002
Editorial changes resulting from December 2001 plenary
Revision 0e – June 1, 2002
Editorial changes resulting from February 2002 plenary.
Revision 0f - August 7, 2002
Editorial changes resulting from June plenary meeting
Revision 0g – August 21, 2002
This major revision implements comments from the line by line review conducted at the plenary meeting

American National Standard
for Information Systems ?

ATA/ATAPI Host Adapters Standard (ATA – Adapter)

Abstract

This standard specifies the Host System Interface used to control AT Attachment Interface devices. It provides a common Programming interface for systems manufacturers, system integrators, software suppliers, and suppliers of intelligent storage devices.

**American
National
Standard**

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer. Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give interpretation on any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

Contents

Tables

Figures

Annexes

Foreword

(This foreword is not part of American National Standard ***-****.)

This standard was developed by the ATA ad hoc working group of Accredited Standards Committee INCITS starting in 2001. This document includes annexes that are informative and are not considered part of the standard.

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, Information Technology Industry Council, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Processing Systems, INCITS. Committee approval of the standard does not necessarily imply that all committee members voted for approval. At the time it approved this standard, the INCITS Committee had the Karen Higginbottom, Chair

(Vacant), Vice-Chair

Monica Vago, Secretary

Organization Represented   Name of Representative

AMP, Inc John Hill, Charles Brill (Alt.)

Apple Computer   David Michael, Jerry Kellenbenz (Alt.)

AT&T Thomas Frost, Paul Bartoli (Alt.)

Bull HN Information Systems, Inc.   Patrick L. Harris

Compaq Computer Corporation   Steven Heil, Seve Park (Alt.)

Eastman Kodak Michael Nier

Hewlett-Packard   Karen Higginbottom,   Donald Loughry (Alt.)

Hitachi America, Ltd.   John Neumann, Kei Yamashita (Alt.)

Hughes Aircraft Company Harold L. Zebrack

IBM Corporation   Ron Silletti, Joel Urman (Alt.)

Institute for Certification of Computer Professionals   Kenneth M. Zemrowski,   Tom Kurihara (Alt.)

Lucent Technologies, Inc. Herbert Bertine, Tom Rutt (Alt.)

National Communications Systems Dennis Bodson, Frack McClelland (Alt.)

National Institute of Standards and Technology   Michael Hogan, Bruce K. Rosen (Alt.)

Panasonic Technologies, Inc..   Judson Hofmann, Terry J. Nelson (Alt.)

Share, Inc.   David Thewlis, Gary Ainsworth (Alt.)

Sony Electronics, Inc. Masataka Ogawa, Michael Deese (Alt.)

Storage Technology Corporation   Joseph S. Zajaczkowski

Sun Microsystems, Inc.   Gary Robinson

Sybase, Inc. Donald Deutsch,   Andrew Eisenberg (Alt.)

Texas Instruments, Inc.   Clyde Camp, Fritz Whittington (Alt.)

Unisys Corporation Arnold F. Winkler, Stephen P. Oksala (Alt.)

U.S. Department of Defense/DISA   Jerry L. Smith, C. J. Pasquariello (Alt.)

U.S. Department of Energy   Carol Blackston,   Bruce R. White (Alt.)

Xerox Corporation   John B. Flannery, Jean Baroness (Alt.)

Subcommittee T13 on ATA Interfaces, that reviewed this standard, had the following members:

Pete McLean, Chairman

Dan Colegrove, Vice Chairman

ATA/ATAPI ad hoc Working Group, that developed this standard, had the following additional participants:

Introduction

This standard encompasses the following:

Clause 1 describes the scope.

Clause 2 provides normative references.

Clause 3 provides definitions, abbreviations, and conventions.

Clause 4 describes ATA host adapters.

Clause 5 describes an ISA Address Decoder Adapter.

Clause 6 describes PCI compatibility and PCI-Native Mode Bus Master Adapters.

Clause 7 Provides requirements for Automatic Direct Memory Access (ADMA) Adapters

American National Standard
for Information Systems -

Information Technology -
ATA/ATAPI Host Adapters Standard (ATA – Adapter)

1 Scope

This standard specifies the AT Attachment Interface between host systems using Automatic Direct Memory Access (ADMA) and storage devices. It provides a common link layer interface for systems manufacturers, system integrators, and software suppliers.

The application environment for the AT Attachment Interface is any host system that has a PCI bus and storage devices contained within the processor enclosure.

This standard maintains a high degree of compatibility with the AT Attachment with Packet Interface – 6 standard (ATA/ATAPI-6), INCITS 361-2002, and while specifying link-layer register definitions and usage information, is not intended to require changes to presently installed devices.

2 Normative References

2.1 Content Imported from Normative Standards

Where material within this standard has been imported from another, normative standard, in whole or in part, it is done so for the sake of readability of this document and the normative source document is identified. The source standard has precedence where there is a difference of definition. References are listed in this section. Each reference has an abbreviated reference form, enclosed within braces '{}'. This abbreviation, braces included, is used in the body of this document. Where the reference cites a standard and working drafts, the cross-reference is to the standard that was current at the time when this document was written.

2.2 Industry Standard References

2.2.1 ANSI Information Technology – AT Attachment with Packet Interface (ATA/ATAPI-5)

Abbreviation: {ATA Standard}

Published standard: Information Technology - AT Attachment with Packet Interface - 5 (ATA/ATAPI-5), ANSI National Standard for Information Systems, # ANSI INCITS 340-2000.

Standard: Information Technology - AT Attachment Interface with Packet Interface Extensions - 6 (ATA/ATAPI-6), INCITS 361-2002

2.2.2 PCI Local Bus Specification

Abbreviation: {PCI Spec}

Standard for the PCI Local Bus, Revision 2.3, published October 19, 2001, by the PCI Special Interest Group (PCI SIG).

This standard is freely available and can be downloaded from www.pcisig.org

2.2.3 PCI Hot-Plug Specification

Abbreviation: {PCI Hot Plug}

Standard for the PCI Local Bus, Revision 1.0, published October 6, 1997, by the PCI Special Interest Group (PCI SIG).

This standard is freely available and can be downloaded from www.pcisig.org

2.2.4 PCI Bus Power Management Specification

Abbreviation: {PCI PMS}

Standard for the PCI Local Bus – Revision 1.1, published December 18, 1998, by the PCI Special Interest Group (PCI SIG).

This standard is freely available and can be downloaded from www.pcisig.org

3 Definitions, Abbreviations, and Conventions

3.1 Definitions and Abbreviations

3.1.1 ADMA

Automatic Direct Memory Access

3.1.2 Adapter

Within the bounds of this standard the term Adapter represents the hardware that is the interface between the ATA host and the ATA Channel. The embodiment of this includes Integrated Circuits, and plug-in adapters.

3.1.3 ADMA Command Chaining

The principle objective of implementing command chaining in the ADMA hardware is to allow the host software and the ADMA hardware to be loosely coupled. To do this, the software can build up a list of tasks (a command chain) for the hardware to execute. The hardware independently reads these requests from host memory and executes the tasks. When the hardware completes a task, it interrupts the host to inform the host that the task is complete, but immediately proceeds to the next task without waiting.

3.1.4 ADMA Mode

An operating mode of the ADMA engine which uses ADMA command chaining.

3.1.5 ADMA Physical Region Descriptor (APRD)

An APRD is an automatic DMA mode data structure that describes areas of host memory that are used during data transfer. (see Clause 7.6.1.8)

3.1.6 ATA Bus

The physical connection between an ATA adapter and an ATA device, that consists of conductors carrying signals.

3.1.7 ATA Bus Release

{ATA Standard} The act of clearing both DRQ and BSY to zero, and setting ATA REL to one, before the action requested by a command is completed. This allows the host to select the other device on the channel. (Applies only to ATA devices that implement Overlap Protocol, by releasing the bus.)

3.1.8 ATA Channel

The ATA Channel is the logical transport mechanism between the ATA host and the ATA devices on an ATA Bus. Each ATA Channel may have up to two ATA devices connected to it.

3.1.9 ATA Command Acceptance

{ATA Standard} Section 3.1.15 "A command is considered accepted whenever the currently selected device has the BSY bit cleared to zero in the Status register and the host writes to the Command register. An exception exists for the EXECUTE DEVICE DIAGNOSTIC and DEVICE RESET commands."

3.1.10 ATA Command Queue (in the device)

{ATA Standard} The ATA Command Queue in the device is the set of all commands that an ATA device has accepted and is currently processing.

3.1.11 ATA device

{ATA Standard} An ATA device is a data storage device. Traditionally, a device on the ATA interface has been a hard drive, but any form of storage device may be placed on the interface, provided that the device adheres to the ATA standard.

3.1.12 ATA DMA

The transfer of data between an ATA device and an ATA adapter under the control of the DMARQ and DMACK signals on the ATA bus. There are two methods of DMA defined: Multiword DMA, where the adapter controls timings, and Ultra DMA, where the sender controls timings.

3.1.13 ATA Host

The ATA host is the host system in which the software that controls the functions of the ATA Subsystem is executed.

3.1.14 ATA Multiword DMA

ATA Multiword DMA is defined to transfer data at up to 16 MB/s. This protocol has traditionally been used in conjunction with the PCI DMA protocol to provide a more efficient means of transferring data through the system.

3.1.15 ATA-n

A shorthand reference to the standard specified in the ATA-n (or ATA/ATAPI-n, as applicable) standards document, whether published as final, circulated in draft form, or only in the planning stage.

3.1.16 ATA Register Mode

ATA Register Mode is an operating mode of the ADMA engine where accesses to the ATA device uses host memory or I/O instructions to access the device registers directly using ATA PIO protocols.

3.1.17 ATA Subsystem

The ATA subsystem includes the ATA hardware elements, which consist of an ATA adapter, an ATA channel, and ATA/ATAPI device(s).

3.1.18 ATA Overlap Protocol

{ATA Standard} The ATA overlap protocol allows an ATA device to perform an ATA Bus Release, so that commands may be executed by another device on the same bus.

3.1.19 ATA Overlapped Command

{ATA Standard} A command is an overlapped command if it is listed as part of the Overlapped or Queued Feature Set.

3.1.20 ATA PIO

{ATA Standard} For the ATA bus, PIO means that data is transferred between the device and the adapter by reading or writing a register in the device. The address of the register and the timing of the transfers are under the control of the adapter.

3.1.21 ATA Ultra DMA

{ATA Standard} ATA Ultra DMA is a high-speed mode of data transfer. The Ultra protocol also defines the use of a CRC to validate that data has been correctly transferred.

3.1.22 ATAPI (AT Attachment Packet Interface) Device

{ATA Standard} An ATA device that implements the Packet Command feature set.

3.1.23 BAR

The Base Address Register (BAR) contains the base addresses of sets of registers accessible through the PCI bus.

3.1.24 Bus Protocol

A bus protocol consists of the sequence of bus signal states, and their timings, which are required in order to transfer commands and data along a bus. There may be more than one protocol available on any one bus.

3.1.25 Byte

A Byte is a unit of data that consists of eight bits of data as described below:

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

3.1.26 Command Complete

Command completion is the completion by the device of the action requested by the command or the termination of the command with an error, the placing of the appropriate error bits in the Error register, the placing of the appropriate status bits in the Status register, the clearing of both BSY and DRQ to zero, and the asserting of INTRQ if nIEN is cleared to zero and the command protocol specifies that INTRQ be asserted.

3.1.27 Command Parameter Block (CPB)

A Command Parameter Block (CPB) is an automatic DMA mode data structure that describes a command to be executed by the ADMA engine. (see clause 7.6.1)

3.1.28 Cyclic Redundancy Check (CRC)

{ATA Standard} Used for the Ultra DMA protocol to check the validity of the data that has been transferred during the last Ultra DMA burst.

3.1.29 DMA

Direct Memory Access. A means of data transfer, between device and host memory, such that host processor intervention is not needed to accomplish the data transfer after initiation of the transfer activity.

3.1.30 DWord

A DWord (Double Word) is a unit of data that consists of four Bytes as represented below:

bit 31   bit 0

Byte 3	Byte 2	Byte 1	Byte 0

3.1.31 Host DMA

{PCI Spec} Host DMA means that data is transferred between the host and the ATA adapter over the PCI bus, using the PCI Burst mode protocol with the adapter as master and the PCI host as target. Once initiated, the transfer requires no host Processor involvement.

3.1.32 PCI

PCI is an acronym for Peripheral Component Interconnect.

3.1.33 Physical Region Descriptor (PRD)

A Physical Region Descriptor (PRD) is a native mode PCI DMA data structure that describes areas of host memory that are used during data transfer. (see Clause 6.7.3)

3.1.34 QWord

A QWord (Quad Word) is a unit of data that consists of eight Bytes as represented below:

bit 63   bit 0

Byte7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0

3.1.35 Word

A Word is a unit of data that consists of two Bytes as represented below:

bit 15   bit 0

Byte 1	Byte 0

3.2 Conventions

3.2.1 Keywords

3.2.1.1 expected: A keyword used to describe the behavior of the hardware or software in the design models assumed by this standard. Other hardware and software design models may also be implemented.

3.2.1.2 mandatory: A keyword indicating items to be implemented as defined by this standard.

3.2.1.3 may: A keyword that indicates flexibility of choice with no implied preference.

3.2.1.4 obsolete: A keyword used to describe bits, bytes, fields, and code values that no longer have consistent meaning or functionality from one implementation to another. However, some degree of functionality may be required for items designated as "obsolete" to provide for backward compatibility. An obsolete bit, byte, field, or command shall never be reclaimed for any other use in any future standard.

Obsolete commands should not be used by the host. Commands defined as obsolete in previous standards may be command aborted by devices conforming to this standard. However, if a device does not command abort an obsolete command, the minimum that is required by the device in response to the command is command completion.

3.2.1.5 optional: A keyword that describes features that are not required by this standard. However, if any optional feature defined by the standard is implemented, the feature shall be implemented in the way defined by the standard.

3.2.1.6 retired: A keyword indicating that the designated bits, bytes, fields, and code values that had been defined in previous standards are not defined in this standard and may be reclaimed for other uses in future standards. If retired bits, bytes, fields, or code values are utilized before they are reclaimed, they shall have the meaning or functionality as described in previous standards.

3.2.1.7 reserved: A keyword indicating reserved bits, bytes, words, fields, and code values that are set aside for future standardization. Their use and interpretation may be specified by future extensions to this or other standards. A reserved bit, byte, word, or field shall be set to zero, or in accordance with a future extension to this standard. The recipient shall not check reserved bits, bytes, words, or fields. Receipt of reserved code values in defined fields shall be treated as a command parameter error and reported by returning command aborted.

3.2.1.8 shall: A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other standard conformant products.

3.2.1.9 should: A keyword indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase "it is recommended".

3.2.2 Precedence

If there is a conflict among text, figures, and tables, the precedence shall be: tables, then figures, and then text.

3.2.3 Names of Registers, Words, Bytes, Bits, Etcetera

The names of registers, words, bytes, bits, and modes begin with uppercase letters. In addition, register names are prefixed with the acronyms PCI, ADMA, or ATA, to indicate which of these register sets they belong to. For example: ATA Status Register, Status Word, Bytes 0-3, and Error Bit.

The names of fields within registers or data structures begin with a lower-case letter followed by upper case letters. The lower case letter allows the reader to identify the structure or register where the field is defined. For example: "aGO" is bit field defined in the ADMA Control Register

3.2.4 Numbers

Numbers are decimal, unless specified otherwise.

Hexadecimal numbers are shown as a string of digits, 0 through 9 or A through F, followed by 'h': e.g., '1AB7h.'

Binary numbers are shown as a string of digits, 0 or 1, followed by 'b': e.g., '10110111b'.'

3.2.5 Nomenclature

b - bits
B - Bytes
G – Giga (10^9)
M – Mega (10^6)
n – nano (10^{-9})
m – milli (10^{-3})
s - second

Page 5

3.2.6 Signal Names

Signal functional names are shown in all uppercase letters. For example: 'CLKRUN'.

3.2.7 Signal States

A signal is 'asserted' when it is driven by an active circuit to the true state. A signal is 'de-asserted' when an active circuit drives it to the false state. A signal is 'released' when it is not actively driven to any state. Some signals have bias circuitry that pulls the signal to either a true state or a false state when no signal driver is actively asserting or de-asserting the signal.

3.2.8 State Diagram Conventions
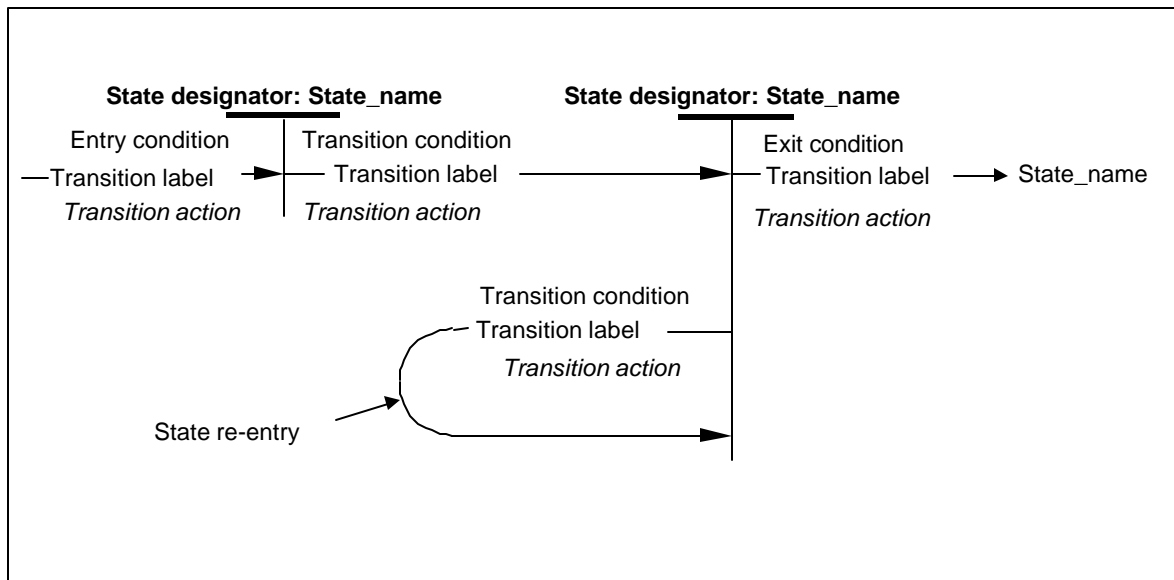
Figure 1 ? State diagram convention

Each state is identified by a state designator and a state name. The state designator is unique among all states in all state diagrams in this document. The state designator consists of a set of letters that are capitalized in the title of the figure containing the state diagram followed by a unique number. The state name is a brief description of the primary action taken during the state, and the same state name may appear in other state diagrams. If the same primary function occurs in other states in the same state diagram, they are designated with a unique letter at the end of the name. Additional actions may be taken while in a state and these actions are described in the state description text.

In device command protocol state diagrams, the state of bits and signals that change state during the execution of this state diagram are shown under the state designator:state_name, and a table is included that shows the state of all bits and signals throughout the state diagram as follows:

    v = bit value changes.
    1 = bit set to one.
    0 = bit cleared to zero.
    x = bit is don't care.
    V = signal changes.
    A = signal is asserted.
    N = signal is negated.
    R = signal is released.
    X = signal is don't care.

Each transition is identified by a transition label and a transition condition. The transition label consists of the state designator of the state from which the transition is being made followed by the state designator of the state to which the transition is being made. In some cases, the transition to enter or exit a state diagram may come from or go to a number of state diagrams, depending on the command being executed. In this case, the state designator is labeled xx. The transition condition is a brief description of the event or condition that causes the transition to occur and may include a transition action, indicated in italics, that is taken when the transition occurs. This action is described fully in the transition description text.

Upon entry to a state, all actions to be executed in that state are executed. If a state is re-entered from itself, all actions to be executed in the state are executed again.

It is assumed that all actions defined in a state are executed within the state and that transitions from state to state are instantaneous.

4 ATA Host Adapters

The ATA interface as defined in {ATA Standard} describes the physical, electrical, timing, protocol and command standards required to transfer data to and from a compliant device. That standard does not differentiate between host adapter and host requirements. The ATA – Adapter standard defines the register and physical requirements of Host adapters. The objective is to enable Host software to be developed that can work with a Host adapter supplied from a variety of vendors.

Host adapters act as a bridge between the Host bus and the ATA bus. Thus Host adapters are required to meet at least two sets of standards. Host software shall be able to configure the adapter for both the Host bus operation and the ATA bus operation. Thus this standard defines, where possible, a common API (Applications Programming Interface) for those functions.

4.1 Adapter Types

The ATA interface has evolved from an original plug in Host adapter on the IBM PC-AT. This adapter controlled hard drives that interfaced to it using an ST506 interface. ATA drives moved the functionality of that adapter from a plug in card into the device. The same register set was retained and thus the most important attribute of the ATA interface was initiated, backward compatibility. Host software and BIOS code did not have to change.

The first ATA Host adapters were address decoder cards plugged into the ISA bus. The cards decoded the I/O addresses of the registers in the ATA register set and connected the ISA bus to the ATA bus. All timings on the ATA bus were those of the ISA bus. As time has progressed the performance of the ATA devices have far exceeded the capabilities of the ISA bus. The majority of Host adapters now reside on the PCI bus and the Host adapters have become more complex involving timing and protocol conversions as a very minimum. An ATA Host adapter can be engineered to work on many host buses. This standard is limited to the original ISA bus and the PCI bus.

4.2 Adapter Modes

4.2.1 Legacy Mode

An adapter is in Legacy Mode when the control of the transfer is through the ATA Command and Control Block Registers including the IO mapped registers. Any data transfers are via PIO mode through the Data register. The addresses of the Command or Control block are configurable in this mode.

4.2.1.1 Compatibility Mode

An adapter is in Compatibility Mode when the control of the transfer is through the ATA Command and Control Block Registers and registers in the adapter. The addresses of the Command or Control block are defined as well as the host's interrupt lines (IRQs). Table 1 defines the four standard I/O address banks.

Table 1 – Compatibility Mode Standard I/O Register Addresses

Channel	Command Block Registers	Control Block Register	IRQ	Alternate IRQ
Primary	IF0h-1F7h	3F6h*	14	None
Secondary	170h-177h	376h*	15	None
Tertiary	1E8h-1EFh	3EEh	11	12 or 9
Quaternary	168h-16Fh	36Eh	10	12 or 9
*NOTE- The Control Block registers were originally defined to include a second register at 3F7h and 377h. This register address was shared with the Floppy Disk adapter on the AT architecture. The floppy drive uses bit 7 of that register; the device used bits 0-6. This register is no longer used in the ATA standard.				

4.2.1.2 PCI-Native Mode

This mode is only applicable to adapters bridging to the PCI bus. In this mode the control of the transfer is through the ATA Command and Control Block Registers and registers in the adapter. The addresses of the Command or Control block are defined in the Base Address Register (BAR) of the adapter and are defined by

the Host software. There is only one Host interrupt line for all the channels attached to an adapter. There is no restriction on the number of adapters there can be in the system.

4.2.2 ADMA Mode

In this mode the ATA Command and Control Block registers are not accessible to the Host. Control is exercised through a data structure held in host memory and adapter registers.

5 ISA Bus Adapter

This type of adapter is commonly called a paddle card for use in PC compatible systems. The function of the adapter is to decode the I/O addresses appropriate to the channels it controls.

5.1 Mode of Operation

Only operates in Compatibility Mode.

5.2 Detection

There is no standard method to detect the presence of this type of adapter. Software may be able to detect the presence of ATA drives by examining the ATA registers at the standard I/O addresses and thereby infer the presence of an adapter.

5.3 Adapter Set Up

There is no standard method used to set up these adapters. In most cases the I/O address banks are set by hard jumpers or by vendor specific registers.

5.4 ATA Bus Timings

ISA timings. No programmable timing is available.

5.5 Electrical and Physical

The electrical and physical specifications of the ATA bus are defined in the {ATA Standard}; the ISA bus characteristics are defined in the {ISA Spec}.

5.6 Registers

The compatibility register set shall be implemented.

6 PCI Compatibility and PCI-Native Mode Bus Master Adapters

PCI Adapters conforming to this standard may operate in Compatibility or Native-PCI Mode. Some adapters can be configured to operate in either mode; some are fixed to one of the modes. The mode configuration may be determined from the PCI Configuration registers.

6.1 Mode of Operation

6.1.1 Compatibility Mode

Adapters operating in Compatibility Mode support two channels conforming to the Primary and Secondary channel address and have a separate IRQ for each channel.

6.1.2 PCI-Native Mode

Adapters operating in Compatibility Mode may support one or two channels.

6.2 Detection

The Class Code fields determine the capabilities.

6.3 Adapter Set Up

The Class Code fields determine the capabilities of an adapter and may be used to configure the channels to Compatibility or PCI-Native mode. The PCI BARs may be used to configure and determine the I/O addresses to use to access the ATA and adapter registers.

6.4 ATA Bus Timings

Determination of the ATA PIO timings, DMA protocols and DMA timings supported is vendor specific. Consequently configuring these attributes is vendor specific.

6.5 Electrical and Physical

The electrical and physical specifications of the ATA bus are defined in the {ATA Standard}; the PCI bus characteristics are defined in the {PCI SPEC} specification.

6.6 PCI Registers

The PCI Adapter implements a subset of the PCI standard type 00h configuration header register set. All registers have the standard meaning as defined in the PCI Specification, Issue 2.2. The registers with specific meanings with respect to this standard are defined below. Register contents that are otherwise defined in the PCI standard are indicates as "PCI". Fields marked 'reserved' are all zeros and read only.

Table 2 – PCI Compatibility and PCI-Native Mode Bus Master Adapters Configuration Registers

Byte Offset	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
00h	PCI			
04h	PCI			
08h	Class Code			PCI
0Ch	PCI			
10h	Base Address 0 -- Base Address of Cmd-Block Regs, ATA Channel X			
14h	Base Address 1 -- Base Address of Control Regs, ATA Channel X			
18h	Base Address 2 -- Base Address of Cmd-Block Regs, ATA Channel Y			
1Ch	Base Address 3 -- Base Address of Control Regs, ATA Channel Y			
20h	Base Address 4 -- Base Address of ATA Bus Master Registers			
24h	Base Address 5 – Vendor Specific			
28h	PCI			
2Ch	Subsystem ID		PCI	
30h	PCI			
34h	PCI			
38h	PCI			
3Ch	PCI			Interrupt Line

6.6.1 PCI Class Code

Address Offset   09h
Size   24 bits

Table 3 – PCI Compatibility and PCI-Native Mode Bus Master Adapters Class Code Registers

Byte Offset	Description	Attribute	Value
09h	Programming Interface Code	See Table 4	See Table 4
0Ah	Sub-class Code	Read Only	01h – IDE
0Bh	Base-Class Code	Read Only	01h – Mass Storage

6.6.1.1 Programming Interface Code

Table 4 defines the usage and values of the Programming and Interface Byte.

Table 4 – PCI Adapter bit definitions in Programming Interface Byte

Bit	Read/write	Description
0	Implementation dependent	Determines the mode that the primary ATA channel is operating in. Clearing this bit to zero corresponds to 'compatibility', setting the bit to one means PCI-native mode. This bit is implemented as read-only if the channel supports only one mode, or read-write if both modes are supported. For implementations that support both modes the power on and hardware reset states are vendor specific.
1	Read Only	This bit indicates whether or not the primary channel has a fixed mode of operation. If this bit is cleared to zero, the mode is fixed and is determined by the read-only value of bit 0. If this bit is set to one, the channel supports both modes and may be set to either mode by writing bit 0.
2	Implementation dependent	Determines the mode that the secondary ATA channel is operating in. Clearing this bit to zero corresponds to 'compatibility', setting the bit to one means PCI-native mode. This bit is implemented as read-only if the channel supports only one mode, or read-write if both modes are supported. For implementations that support both modes the power on and hardware reset states are vendor specific.
3	Read Only	This bit indicates whether or not the secondary channel has a fixed mode of operation. If this bit is cleared to zero, the mode is fixed and is determined by the read-only value of bit 2. If this bit is set to one, the channel supports both modes and may be set to either mode by writing bit 2.
4-6	Read Only	Reserved and shall be cleared to zero.
7	Read Only	This bit shall indicate that the adapter can do bus master operation when it is set to one. This bit shall indicate that the adapter is not capable of bus master operation when it is cleared to 0.

6.6.2 PCI Base Address Registers (BAR)

Base Address Registers 0-3 have Bit zero hard-wired to one to indicate I/O space. Initialization of the following BARs is described in the {PCI Specification}

6.6.2.1 PCI Base Address Register (BAR) 0
This is the base address for the command block registers for ATA Channel X.
Address Offset   10h
Default Value   00000001h

When operating in Compatibility Mode any write to the BAR shall be ignored and the value 1F0h shall always be used. In Compatibility Mode an independent IRQ shall be provided that is connected to IRQ14. When the adapter is disabled by using the I/O Enable bit in the PCI Command register, the adapter shall not respond to any I/O addresses, and shall release its IRQ connections.

Attribute   Bits 31-16 may be Read Only, Bits 15-3 Read/Write, Bits 2-0 Read Only
Size   32 bits

6.6.2.2 PCI Base Address Register (BAR) 1
This is the base address for the control register for ATA Channel X. Note that because of the Dword alignment of PCI, the Device Control and Alternate Status Registers are at offset 02h from this base.
For example, to put those registers at address 3F6h, this register shall be set to 3F4h (+ Bit 0).
Address Offset   14h
Default Value   00000001h
When operating in Compatibility Mode any write to the BAR shall be ignored and the value 3F5h always be used.
Attribute   Bits 31-16 may be Read Only, Bits 15-2 Read/Write, Bits 1-0 Read Only.
Size   32 bits

6.6.2.3 PCI Base Address Register (BAR) 2
If the device implements two channels this is the base address for the command block registers for ATA Channel Y. If the device only supports one channel this base address is read only and cleared to zero,
Address Offset   18h
Default Value   00000001h
When operating in Compatibility Mode any write to the BAR shall be ignored and the value 170h shall always be used. In Compatibility Mode an independent IRQ shall be provided that is connected to IRQ15. When the adapter is disabled by using the I/O Enable bit in the PCI Command register, the adapter shall not respond to any I/O addresses, and shall release its IRQ connections
Attribute   Bits 31-16 may be Read Only; Bits15-3 Read/Write; Bits 2-0 Read Only.
Size   32 bits

6.6.2.4 PCI Base Address Register (BAR) 3
If the device implements two channels this is the base address for the control registers for ATA Channel Y. If the device only supports one channel this base address is read only and cleared to zero. Note that because of the Dword alignment of PCI, the Device Control and Alternate Status Registers are at offset 02h from this base.
Address Offset   1Ch
Default Value   00000001h
When operating in Compatibility Mode any write to the BAR shall be ignored and the value 376h shall always be used.
Attribute   Bits 31-16 may be Read Only, Bits 15-2 Read/Write, Bits 1-0 Read Only.
Size   32 bits

6.6.2.5 PCI Base Address Register (BAR) 4
Base address of the ATA Bus Master I/O registers.
Address Offset   20h
Default Value   00000001h
Attribute   Bits 31-16 may be Read Only, Bits 15-3 Read/Write, Bits 2-0 Read Only.
Size   32 bits

6.6.2.6 PCI Base Address Register (BAR) 5

This is a vendor specific BAR

6.6.3 PCI Interrupt Line

The host BIOS and host software may use this location to store the system interrupt (IRQ) allocated to this device. The adapter does not use this information. BIOS and host software may use this location to store the information.
Address Offset   3Ch
Default Value   00h

6.7 ATA Bus Master Registers

The bus master -ATA function uses 16 bytes of I/O space. All bus master -ATA I/O space registers can be accessed as byte, word, or Dword quantities. The base address for these registers is PCI BAR 4. The description of the 16 bytes of I/O registers follows:

Table 5 – ATA Bus Master Register Offsets

Offset from Base Address	Register	RegisterAccess (1,2)
00h	ATA Bus Master Command register Primary	R/W
01h	Device Specific	
02h	ATA Bus Master Status register Primary	RWC
03h	Device Specific	
04h-07h	ATA Bus Master PRD Table Address Primary	R/W
08h	ATA Bus Master Command register Secondary	R/W
09h	Device Specific	
0Ah	ATA Bus Master Status register Secondary	RWC
0Bh	Device Specific	
0Ch-0Fh	ATA Bus Master PRD Table Address Secondary	R/W

Notes:

1) R/W means that the bit may be both written and read.

2) RWC means that the bit may be read. If set it is cleared by writing a 1

6.7.1 ATA Bus Master Command Register

Register Name: ATA Bus Master Command Register

Address Offset: Primary Channel: Base + 00h

Secondary Channel: Base + 08h (for adapters that do not support this channel these registers shall return zero when read).

Default Value: 00h

Attribute: Read / Write

Size: 8 bits

Table 6 – ATA Bus Master Command Register

Bit	Description.
7-4	Shall return zero on reads.
3	**Read or Write Control:** This bit sets the direction of the bus master transfer: when cleared to zero, PCI bus master reads are performed. When set to one, PCI bus master writes are performed. This bit shall NOT be changed when the bus master function is active.
2-1	**Shall** return zero on reads.
0	**Start/Stop Bus Master:** Bus master operation of the adapter is enabled by setting this bit to one. Bus master operation begins when this bit is detected changing from a zero to a one. The adapter shall only transfer data between the ATA device and host memory when this bit is set to one. Master operation may be halted by clearing this bit to zero. All state information is lost when a zero is written; Master mode operation cannot be stopped and then resumed. If this bit is cleared to zero while bus master operation is still active the bus master command is aborted and data transferred from the device may be discarded before being written to host memory. In any event the adapter should terminate any DMA burst in progress compliant with the ATA protocol being used. If the UDMA protocol is being used there may be a CRC error report. The ATA device may be in an indeterminate state. This bit is intended to be cleared to zero after the data transfer is completed, as indicated by either the Bus Master ATA Active bit or the Interrupt bit of the Bus Master ATA Status register for that ATA channel being set to one, or both.

6.7.2 Adapter Bus Master Status Register

Register Name: Bus Master ATA Status Register

Address Offset: Primary Channel: Base + 02h

                    Secondary Channel: Base + 0Ah (for adapters that do not support this channel these registers shall return zero when read).

Default Value: 00h

Attribute: Read/Write Clear

Size: 8 bits

Table 7 – Bus Master ATA Status Register

Bit	Description
7	**Simplex only:** This read-only bit indicates whether or not both bus master channels (primary and secondary) can be operated at the same time. If the bit is a zero, then the channels operate independently and can be used at the same time. If the bit is a one, then only one channel may be used at a time. **NOTE -** Some vendors use this bit in a vendor specific manner when special features are enabled.
6	**Device 1 DMA Capable:** This read/write bit is set by device dependent code (BIOS or host software) to indicate that Device one for this channel is capable of DMA transfers, and that the adapter has been initialized for optimum performance.
5	**Device 0 DMA Capable:** This read/write bit is set by device dependent code (BIOS or host software) to indicate that Device zero for this channel is capable of DMA transfers, and that the adapter has been initialized for optimum performance.
4-3	**Reserved.** Must return zero on reads.

2	**Interrupt:** This bit shall be set to one by the rising edge of the ATA channel's interrupt line (INTRQ). This bit shall be cleared to zero when a one is written to it by software. Software can use this bit to determine if an ATA device has asserted its interrupt line (see ATA standard). When this bit is read as a one, all data may have been transferred to the Host's system memory. The adapter shall not set this bit to one until it has flushed any internal data buffers. NOTE - even if the device status is read, causing the device to deassert INTRQ, this bit shall be cleared as well.
1	**Error:** This bit is set when the adapter encounters an error in transferring data to/from host memory. The exact error condition is bus specific and can be determined in a bus specific manner. This bit is cleared when a one is written to it by software.
0	**Bus Master -ATA Active:** This bit shall be set to one when the Start bit is written to the ATA Bus Master Command registers. This bit shall be cleared to zero when the last transfer for a region is performed, where EOT for that region is set in the region descriptor. It is also cleared when the Start bit is cleared in the ATA Bus Master Command register. When this bit is read as a zero, all data transferred from the device during the previous bus master command has been transferred to the Host's system memory, unless the bus master command was aborted. When this bit is zero all data has been written to the device

6.7.3 Physical Region Descriptor (PRD) Table Pointer Register

Register Name: Descriptor Table Pointer Register

Address Offset: Primary Channel: Base + 04h

                                  Secondary Channel: Base + 0Ch (for adapters that do not support this channel these registers shall return zero when read).

Default Value: 00000000h

Attribute: Read / Write

Size: 32 bits

Table 8 – PRD Table Pointer Register

Bit	Description
31-2	Base address of Descriptor table. Corresponds to A[31-2]
1-0	Reserved

The PRD Table shall be Dword aligned. The Descriptor Table shall not cross a 65,536 (64KB) boundary in host memory.

6.8 Interrupt Line Considerations

When a channel is in Compatibility Mode the IRQ used by the channel shall be the compatibility IRQ. PCI interrupt lines shall not be affected by that channel's interrupt. Conversely, when the channel is in PCI-native mode the interrupt signals from both channels shall be connected to the appropriate interrupt pin number (INTA). Compatibility IRQs shall not be affected and if connected shall be released.

Connections of channel interrupt signals to the compatibility IRQs should be released until the adapter is enabled via the PCI Command register in PCI Configuration Space. The adapter is enabled when a one is written to the I/O enable bit (bit 0) in the Command register.

6.9 Bus Master Operation

When transferring data to or from an ATA device using an ATA DMA protocol the adapter uses PCI bus master protocols to transfers the data to or from the Host's memory.

The Bus Master mode-programming interface is an extension of the standard ATA programming model. This means that devices can always be dealt with using the standard ATA programming model, with the Bus Master mode functionality used when the appropriate software and devices are present. Bus Master operation is designed to work with any ATA device that supports DMA transfers on the ATA bus. Devices that only work in PIO mode can be used through the standard ATA programming model.

The programming interface defines a simple scatter/gather mechanism allowing large transfer blocks to be transferred to and from non-contiguous areas of host memory.

Master ATA Adapters shall default upon PCI reset to Mode zero Multiword DMA timings. This ensures operation with DMA capable ATA devices without the need for adapter vendor-specific code to initialize adapter-specific timing parameters.

6.9.1 Physical Region Descriptor (PRD) Table

Before the start bit is set in the Bus Master register, it is given a pointer to a valid PRD Table This table contains some number of Physical Region Descriptors (PRDs), which describe areas of host memory that are involved in the data transfer. The descriptor table shall be aligned on a Dword boundary and the table cannot cross a 65,536 (64KB) Byte boundary in host memory.

6.9.2 Physical Region Descriptor

The physical host memory region to be transferred is described by a Physical Region Descriptor (PRD). The data transfer will proceed sequentially from the first PRD until all regions described by the PRDs in the table have been transferred.

Each PRD entry is 8 bytes in length. The first 4 bytes specify the byte address of a physical host memory region. The next two bytes specify the count of the region in bytes (65,536 Byte limit per region). A value of zero in these two bytes indicates 65,536. Bit 7 (EOT) of the last byte indicates the end of the table; bus master operation terminates when the last descriptor has been retired.

Table 9 – Physical Region Descriptor Table Entry

	Byte 3	Byte 2	Byte 1	Byte 0	
Dword 0	Host memory Region Physical Base Address [32-1]				0
Dword 1	EOT	Vendor Specific	Byte Count [15-1]		0

The host memory region specified by the descriptor is further restricted such that the region shall not cross a 65,536 boundary. The sum of the descriptor byte counts must be equal to, or greater than the size of the device transfer request. The host software shall terminate the bus master transaction by resetting bit zero of the ATA Bus Master Command register to zero.

The adapter may update the Byte Count during its operation. Host software should not assume that the PRD has been updated and should re-initialize its contents before reusing the structure.

6.9.3 Standard Programming Sequence

To initiate a bus master transfer between host memory and an ATA DMA device, the following steps are required:

1. Software prepares a PRD Table in host memory.

2. Software provides the starting address of the PRD Table by loading the PRD Table Pointer Register. Setting of the Read/Write Control bit specifies the direction of the data transfer. Clearing the Interrupt and Error bits in the ATA Bus Master Status register to zero readies the adapter for a data transfer.

3. Software issues the appropriate DMA transfer command to the device.

4. Software initiates the bus master function by writing a one to the Start bit in the ATA Bus Master Command Register for the appropriate channel.

5. The adapter transfers data to/from host memory responding to DMA requests from the ATA device.

6. At the end of the transfer the ATA device asserts an interrupt if nIEN is cleared to zero.

7. For transfers from the device to the host the adapter shall first transfer any internal data buffers before first setting the interrupt bit and asserting the Host interrupt signal to host memory.

8. In response to the interrupt, software resets the Start/Stop and interrupt bits in the ATA Bus Master Command register. The software then reads the adapter's and device's Status registers to determine if the transfer completed successfully.

9. If there is less data transferred than is specified by the PRD table see Table 10 for appropriate action.

6.9.4 ATA Bus Master Status Register Bit Interpretation

The table below gives a description of how to interpret the Interrupt and Active bits in the Adapter status register after a DMA transfer has been started.

Table 10 – Adapter Bus Master Status Register bits

Interrupt	Error	Active	Description:
0	0	1	DMA transfer is in progress and the ATA device has not asserted INTRQ.
1	0	0	The ATA device asserted INTRQ and the adapter exhausted the PRDs. This is the normal completion case where the size of the physical host memory regions was equal to the ATA device transfer size.
1	0	1	The ATA device asserted INTRQ but the adapter has not reached the end of the physical host memory regions defined in the PRDs. This is a valid completion case where the size of the physical host memory regions was larger than the ATA device transfer size.
0	0	0	This bit combination signals an anomaly. The PRD's specified a smaller size than the ATA transfer size and as a result the device may have to be reset. See clause 6.9.5.
X	1	X	The adapter has some problem transferring data to/from host memory. Specifics of the error have to be determined using bus-specific information. A timeout may be required to detect this condition.

6.9.5 Error Conditions

If the adapter encounters an error while doing the bus master transfers it shall stop the transfer (i.e. reset the Active bit in the adapter Bus Master Command register) and set the Error bit in the adapter Bus Master Status register. The adapter does not generate an interrupt when this happens. The host software may use device or adapter specific information (e.g.; PCI Configuration Space Status register) to determine what caused the error. In any event the adapter should terminate any DMA burst in progress compliant with the ATA protocol being used. If the UDMA protocol is being used there may be a CRC error to report. The ATA device may be in an indeterminate state.

7 Automatic Direct Memory Access (ADMA) Adaptors - General Description

7.1 Background

The performance of ATA devices has increased dramatically over time, including the introduction of the Overlapped and Queued Feature Sets. However, systems using these faster devices have not shown all of the expected benefits. This limited performance improvement is a consequence of the demands the PCI compatibility and PCI-native mode bus master adapters places on the resources of the host. The problem is amplified by the design of the standard ATA host software used in some operating systems. This software effectively treats ATA transfers as a single-threaded entity, with correspondingly long latencies to fully service the interrupts. The Automatic DMA (ADMA) engine is designed to address these problems and add features that make it, and ATA devices, suitable for true multi-threading applications. This host adapter shall not support single or multiword DMA.

7.2 The ADMA Engine

The objective of the ADMA design is to drastically increase the performance of systems that use ATA devices.

To fully optimize the system throughput, the ADMA engine implements a command chaining technique to de-couple the host command sequence from the channel execution. This decoupling is accomplished by having the traditional host software-to-device I/O negotiation executed in hardware. This allows true multitasking and the ability to effectively exploit the Overlapped and Queued Feature Sets.

The ADMA engine works in two modes, termed ATA Register and ADMA

7.2.1 Modes of Operation

In ATA Register Mode, the host software directly writes/reads the ATA registers. The only function performed by the ADMA engine is to obey the Programmed I/O (PIO) timing rules for the current PIO mode. In this mode, the ATA interrupts are routed directly to a PCI interrupt.

In ADMA Mode, the ADMA engine controls all aspects of the ATA protocols including interception and routing of the ATA interrupt as appropriate.

7.2.2 PCI Bus

The PCI bus conforms to the {PCI SPEC}, the {PCI PMS}, and the {PCI Hot Plug} requirements. The ADMA contains configuration header registers consistent with its implementation as a single PCI function device, and thus drives a single interrupt line, see 7.2.6.

7.2.3 PCI Configuration

The ADMA may be implemented to support one or two ATA channels. The PCI configuration header registers contain one Memory Mapped Base Address Register (BAR) associated with the ADMA and ATA Registers, two I/O BARs for each ATA channel, and an Expansion ROM BAR associated with the optional BIOS.

7.2.4 FIFOs and FIFO Control (optional)

Each ADMA channel contains a FIFO used to buffer data during transfers. Each FIFO shall be a power of two Quad-Words (Qwords) long and is controlled by two host programmable registers: the ADMA FIFO Input Threshold (FITR) and ADMA FIFO Output Threshold (FOTR). These registers are used by to control the FIFO by indicating the burst size to/from the PCI bus The ADMA FIFO Input Threshold controls the requested PCI data transfer burst length from the ADMA to the host, while the ADMA FIFO Output Threshold controls the requested PCI data transfer burst length from the host to the ADMA. Adjusting the FIFO thresholds controls the length of the PCI transaction.

7.2.5 ATA Bus Transfer Modes

The ATA PIO protocol shall be automated in the ADMA. This means that the ADMA executes PIO transfers to the device while performing PCI Burst mode transfers on the PCI bus. This method of operation is termed "DMA-Assisted PIO mode". The ADMA uses DMA-Assisted PIO mode for devices not implementing Ultra DMA.

In ATA Register Mode, the ADMA supports data transfer to/from the ATA bus using PIO protocols.

In ADMA mode, the ADMA uses Ultra DMA between the device and the ADMA and uses PCI Burst mode DMA between the ADMA and the host.

7.2.6 ATA Interrupts

The ADMA relies on the use of the ATA device interrupt INTRQ. Under no circumstances shall the host software set nIEN to one, thereby disabling the ATA INTRQ signal. The ADMA engine is interrupt driven and may not operate correctly in ADMA Mode with this bit set to one.

7.2.7 ADMA Interrupts

The ADMA generates the ADMA interrupt signal from three different sources of interrupt:

1. ADMA Interrupts. The ADMA Interrupt signal is asserted by the ADMA to inform the host software of events occurring within the ADMA engine.

2. Unsolicited Interrupts. For systems that require an external interrupt the ADMA Unsolicited Interrupt (UIRQ) signal shall be available.

3. ATA Device Interrupts. When in ATA Register Mode, the ATA device interrupt is the ATA INTRQ signal coming from the ATA/ATAPI bus.

The above signals from each channel are OR-ed to generate the Channel X Interrupt and the Channel Y Interrupt. These signals are OR-ed in turn to generate the PCI interrupt signal.

7.3 ADMA Overview

The features of the ADMA engine are designed to provide access to the ATA registers using PIO reads and writes. Many new features drastically improve performance as well as functionality. This section briefly introduces these features. For this reason, ADMA shall support both IO and Memory mapped operation.

7.3.1 Single Stepping and Continuous DMA

The ADMA may be implemented using either Single Stepping DMA or Continuous DMA. The single stepping DMA version read and processes one command at a time, interrupting the host after each command is complete and waits for the host to instruct it to process the next command. The continuous DMA version does not stop between commands and these commands are processed without host intervention.

7.3.2 Accessing ATA and ADMA Registers

The ATA device's Command and Control Registers are accessible through the ADMA by either I/O or memory mapped registers. The control of the ADMA engine is through a set of memory mapped registers.

7.3.3 ADMA Control Operating Modes

Two modes of operation are implemented: ATA Register Mode, and ADMA Mode.

7.3.3.1 ATA Register Mode

ATA Register Mode is the power-on and reset default. In this mode, the ATA adapter acts as an address decoder for the host. All reads and writes are performed using host I/O or host memory instructions. The only function performed by the ADMA is to control the signal timings of the ATA bus and to respond to PCI signals. In this mode, all data transfers use the PIO protocols, and ATA bus interrupts are directly mapped onto the PCI interrupt signal. Note that both channels X and Y use the single PCI interrupt signal.

7.3.3.2 Automatic DMA Mode

The ADMA autonomously follows a command chain of Command Parameter Blocks (CPBs) mapped in host memory as described in Section 7.3.5. In this mode, data transfers can be either Ultra DMA or DMA-assisted PIO.

In ADMA Mode, the ATA legacy registers are not available. In this mode, any access to the ATA Status (or ATA Alternate Status) register returns a value with bit 7 (BSY) set to one with all other bits cleared to zero. Any read to other registers returns an indeterminate value. Any write to an ATA register shall be ignored.

7.3.4 ADMA Registers

7.3.4.1 ADMA Control (ADMCTL)

The ADMA Control Register contains bits to control the mode and flow of operation.

7.3.4.2 ADMA Status (ADMSTAT)

The ADMA Status Register contains bits that indicate the current state of the process and interrupt reason.

7.3.4.3 CPB Search Count (CCNT)

The ADMA uses this value to determine when to stop processing CPBs. This value is usually set to the number of entries in the CPB chain.

7.3.4.4 Current CPB Address (CCPB)

The Current CPB Address Register contains the 32-bit address of the CPB currently being processed.

7.3.4.5 Next CPB Address (NCPB)

The ADMA Next CPB Address Register contains the 32-bit address of the next CPB to be processed by the ADMA. The host software shall initialize the contents of this register before entering ADMA Mode. The ADMA updates this register as it progresses through the CPB chain, except when accessing the CPB Lookup Table.

7.3.4.6 CPB Lookup Table Address Register (CPBLAR)

The CPB Lookup Address Register contains the 32-bit address of the base of the CPB Lookup Table. If the host software is using overlapped/queued commands, it shall initialize the contents of this register before entering ADMA Mode. The ADMA uses this address to locate the correct CPB when servicing a released overlapped/queued ATA or Packet Command.

7.3.4.7 ADMA FIFO Input Threshold Register (FITR) (optional)

FITR controls the requested PCI data transfer burst length from the ADMA to the host.

7.3.4.8 ADMA FIFO Output Threshold Register (FOTR) (optional)

FOTR controls the requested PCI data transfer burst length from the host to the ADMA.

7.3.5 ADMA Data Structures

The ADMA requires the following data structures to control and manage ATA devices when in ADMA Mode. These data structures shall be physically located in host memory with each entry of the data structure in physically contiguous space. Some HBA's require that all addresses be Qword aligned.

7.3.5.1 CPB Chain

The CPB chain may be a circular linked-list of CPB entries. A CPB is a data structure used during ADMA Mode to store parameters that control the ADMA engine, and define an ATA command for the ATA device. A chain of CPBs is created in host memory as a linked-list with each CPB pointing to the next CPB. Within each CPB, there is a pointer to a APRD chain. An APRD chain defines the host memory locations where data is to be written to/read from. Figure 2 illustrates the CPB structure and APRD relationship.
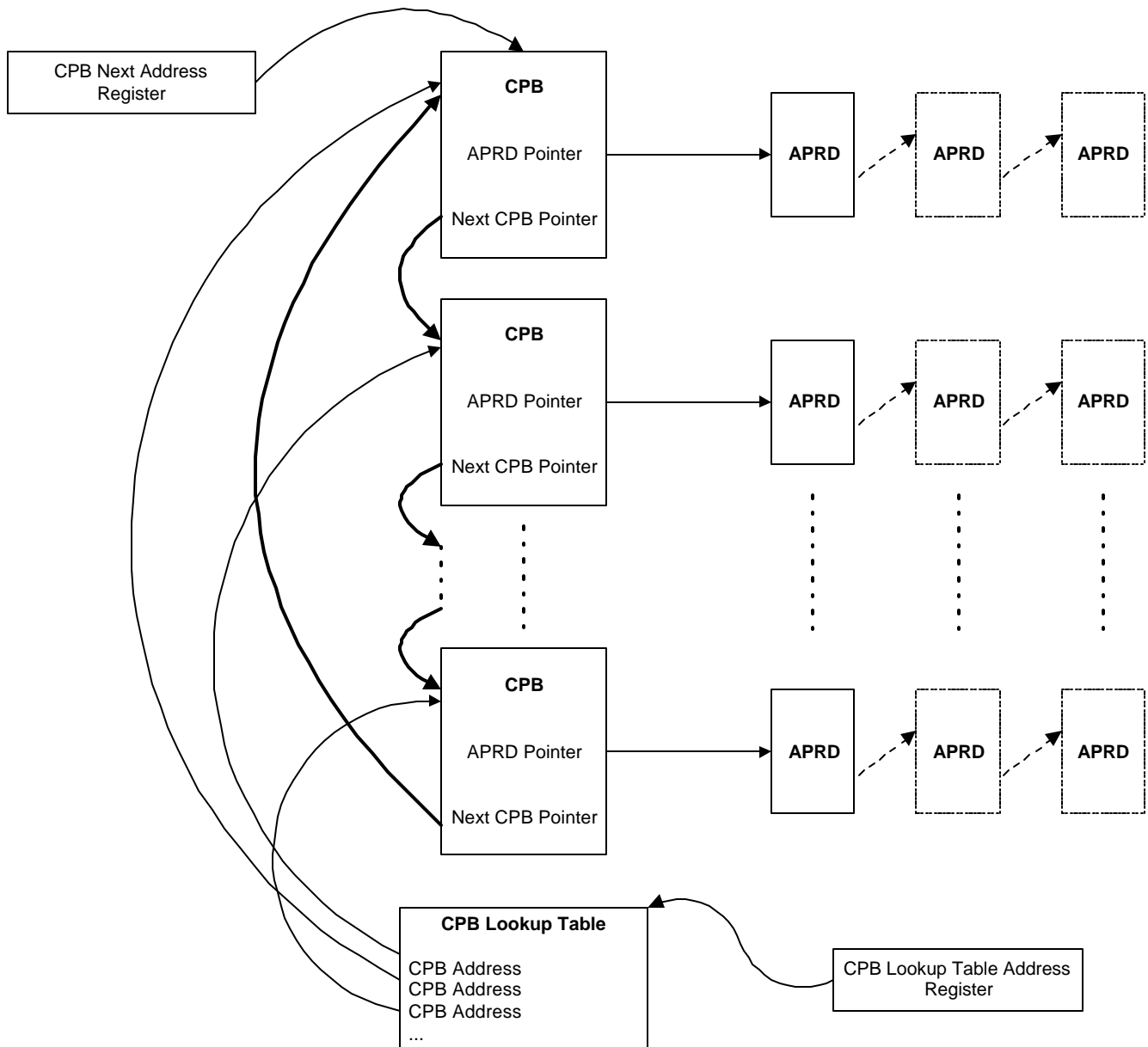
Figure 2 – ADMA Data Structures

An individual CPB can be in one of four States: Not-Valid, Valid-Waiting, Valid-Processing or Released. A CPB that is Not-Valid is under the control of the host. A CPB that is Valid-Waiting, Valid-Processing, or Released is under the control of the ADMA and should not be updated by the host, except for the cVLD bit (see 7.6.3).

7.3.5.2 APRD Chain

An APRD chain is a linked-list of one or more APRD entries used to transfer data for a single CPB. This is a superset of the compatible mode PRD pointers described in 6.9.1. Each entry contains the physical location to be used as the source or destination of the current transfer, and the transfer length. The transfer location may be a contiguous host memory area or an I/O address. Each entry also contains a pointer to the next entry, as well as values to control the method and mode of the transfer.

7.3.5.3 CPB Lookup Table

The host constructs a CPB Lookup Table in host memory for use in overlapped/queued operation. It is pointed to by the CPB Lookup Address Register, which is initialized by the host software. Each entry in the table is a Qword holding the physical address of a CPB. Figure 2 illustrates the CPB Lookup Table.

7.3.5.4 Data Structure Initialization

A valid APRD chain shall be constructed. An APRD chain defines the host memory locations where data is to be written to/read from. A valid CPB chain shall be constructed consisting of one or more CPB entries with the Next CPB field pointing to the physical host memory address of the next CPB in the chain (the Next CPB field in a chain of one CPB would point to itself). Each CPB shall be initialized to point to the head of an APRD chain. All CPBs shall be set to Not-Valid. The CPB Lookup Table shall be constructed. The address of the first CPB shall be written into the ADMA Next CPB Address Register, and the base of the CPB Lookup Table shall be written into the ADMA Lookup Table Address Register.

7.3.6 ADMA Engine Initialization

Before entering ADMA Mode, the host writes a CPB address into the ADMA Next CPB Address Register, and writes into the CPB Search Count Register the number of CPBs to scan before stopping. The value in the CPB Search Count Register is used by the ADMA to refresh an internal counter. This internal counter is refreshed each time the host writes a one to aGO indicating that the ADMA should begin processing the CPB chain. The ADMA decrements this internal count each time a CPB is encountered. When this internal counter reaches zero, the ADMA stops fetching CPB entries. The value set in the CPB Search Count Register determines the number of CPBs that will be accessed after aGO is written as one.

7.3.7 Time-outs

The ADMA shall not automatically time out an ATA command. The host software is responsible for timing out commands. Status information in the CPB chain and the ADMA Registers is available for host software to be able to determine the status of the ADMA. Host software is able to pause the ADMA in the event of a command time-out, revert to ATA Register Mode, and directly address the device registers.

7.3.8 Non-Queued Operation

The host software assembles CPB entries in the host's memory and indicates that the ADMA should examine the CPB chain by writing aGO to one in the ADMA control register. Starting from the current value in the ADMA Next CPB Address Register, the ADMA Sequencer reads the current CPB using a PCI master mode burst.

1. If the CPB is ready to be processed, then the ADMA proceeds to step 2. If the CPB is not ready to be processed, the next CPB is read until a valid entry is found or the ADMA internal CPB counter has decremented to zero.

2. The contents of the CPB ATA command block are read by the ADMA.

3. The ADMA updates the Next CPB Address Register from the current CPB to point to the next CPB in the chain

4. The ADMA stores the start address of the APRD chain

5. If the CPB involves a data transfer, the ADMA reads the first APRD entry, using a PCI master mode burst. The data address and transfer count in this entry are used by the ADMA to control the data transfer. The control information in this entry is used by the ADMA to manage the transfer on the ATA Bus.

6. The ADMA writes the ATA device's registers with parameters and the command.

7. If the ATA data transfer is via PIO, the ADMA monitors the ATA Status Register and the ATA INTRQ signal, to determine when the device is ready to transfer data.

8. If the transfer is via Ultra DMA, the ATA device indicates that it is ready, by asserting the ATA signal DMARQ.

9. When all the data has been transferred as indicated by the APRD count, the ADMA accesses the next APRD entry from the address located in the current APRD. This process is repeated until the total transfer is complete.

10. At the conclusion of a CPB, the ADMA updates the CPB with status information. The PCI Interrupt signal is then asserted if the CPB indicates a request for an interrupt on completion of the CPB, or if an ATA error occurs.

11. In the single step version the ADMA engine shall go into its idle state and wait for the host to write aGO as one before accessing the next CPB at step 1.

12. In the continuous version the ADMA shall continue to step 1

7.3.9 Queued Operation

The overlapped/queued protocols differ for ATA and ATAPI devices. Only certain commands can be overlapped or queued. On each such command, there is a flag in the CPB that indicates that it contains a queued or overlapped ATA command. Overlapped/queued commands shall not be mixed with non-queued commands for the same device within a CPB chain.

a) The sequence commences as in section 7.3.8. After step 6 in that sequence when the Queued command is written to the ATA device, the ADMA waits for the DMARQ signal to be asserted, or ATA BSY to be cleared to zero.

b) If an ATA device is ready to transfer data, it asserts DMARQ and the transfer completes. If the device is not ready to transfer data for the current command, it may release the ATA bus, by clearing BSY to zero and with the ATA Release Bit (REL) set to one. In either case, the ATA Service Bit (SERV) may be set to one, to indicate that the device is ready to transfer data for a previously queued command.

c) If the ATA INTRQ signal is asserted and ATA REL is set to one and ATA SERV is cleared to zero, then the ADMA sets the current CPB to the Released State and proceeds to step 1 in 7.3.8 without asserting the PCI Interrupt signal. The next valid CPB is executed, by writing the command to the specified ATA device, thereby creating a queue of commands in one or both ATA devices.

d) If ATA SERV is set to one, the ADMA writes an ATA Service command to the device, and then polls the ATA status Register. When ATA DRQ is set to one, the ADMA reads the TAG from the ATA device, reads the ATA DEV Bit to determine the current device, and combines the TAG, DEV, and CPBLAR to fetch the Released CPB's address from the CPB Lookup Table. The ADMA then performs the data transfer as in the non-queued case.

e) When the ATA bus is in the IDLE State, auto-polling is enabled and queues have been built in both devices on a channel the ADMA alternately selects each device (auto-poll). This enables a device to assert the ATA INTRQ signal to indicate that it requires service. The ADMA stops the auto-poll sequence if the ATA INTRQ signal is asserted and reverts to step d.

7.3.10 Host Control of Ultra DMA Burst Length

ADMA provides the host with the ability to control the size of an Ultra DMA data burst. The device may choose to provide a smaller burst than requested by the host. Ultra DMA mode transfers include the use of a Cyclic Redundancy Check (CRC). The CRC is calculated over the entire block of data transferred. A CRC was introduced into the Ultra DMA protocol to increase the integrity of the data on the ATA cable. The effectiveness of a CRC reduces as the length of the transfer increases. The ADMA has the ability to break each transfer into smaller units by terminating the burst after transfer of a host-specified number of Sectors. The maximum size of each burst is controlled by the APRD for that block of data.

7.4 ADMA PCI Registers

7.4.1 PCI Configuration Header Registers

All ADMA PCI registers have the standard meaning as defined in the PCI Specification, Issue 2.2. The ADMA implements a subset of the standard type 00h configuration header register set. The implemented registers, and device-specific values, are described below. The contents of fields marked 'Reserved' shall be preserved.

Table 11 – ADMA PCI Configuration Space Header Registers

Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0	Byte Offset
PCI Device ID		PCI Vendor ID		00h
PCI Status Register		PCI Command Register		04h
PCI Class Code			PCI IC Revision	08h
Reserved	PCI Header Type	PCI Latency Timer	PCI Cache Line Size	0Ch
Base Address 0 – Base Address of Command-Block Registers,			ATA Channel X	10h
Base Address 1 – Base Address of Control Registers,			ATA Channel X	14h
Base Address 2 – Base Address of Command-Block Registers,			ATA Channel Y	18h
Base Address 3 – Base Address of Control Registers,			ATA Channel Y	1Ch
Base Address 4/5 – Base Address of Memory Mapped ATA Channel and ADMA Registers				20h-27h
Reserved				28h
PCI Subsystem ID		PCI Subsystem Vendor ID		2Ch
PCI Expansion ROM Base Address				30h
Reserved			PCI Capability Ptr.	34h
Reserved				38h
PCI Max. Latency	PCI Min. Grant	PCI Interrupt Pin	PCI Interrupt Line	3Ch
Vendor Specific				…
Power Management Capability		Ptr. to Nxt Capability	Capability ID	50h
Data Register	Bridge Support Ext.	Power Management Control/Status		54h
Vendor Specific				58-FFh

7.4.1.1 PCI Vendor ID

Address offset   00h

Value   xxxxh (PCI Vendor ID)

Attribute   Read Only

Size   16 bits

7.4.1.2 PCI Device ID

Indicates the implemention ID of the ADMA device.

Address offset   02h

Value   xxxxh (Vendor Specific)

Attribute   Read Only

Size   16 bits

7.4.1.3 PCI Command Register

The value in this register is set by the host to enable various PCI functions. Default on reset is everything disabled.

Address Offset   04h

Default Value   0000h

Attribute   Read/Write

Size   16 bits

Table 12 – ADMA PCI Command Register

Bit	Attribute	Description
0	R/W	Target I/O enable. A value of zero disables Base Address Registers 0 – 4.
1	R/W	Memory Space Enable. A value of zero disables the Expansion ROM Base Address Register and the Memory Base Address Register.
2	R/W	Master Enable. A value of zero disables the Master mode function of the ADMA.
3	R	Reserved.
4	R/W	Memory Write and Invalidate Enable. A value of zero disables the function in the ADMA.
5	R	Reserved.
6	R/W	Parity Check Enable. A value of zero causes parity errors to be ignored.
15-7	R	Reserved.

7.4.1.4 PCI Status Register

Provides status information related to PCI bus events. Bits indicated as "clear" are cleared by writing a one to that bit position.

Address Offset   06h

Default Value   See Table Below

Attribute   Read Only/Clear

Size   16 bits

Table 13 – ADMA PCI Status Register

Bit	Description	Default	Fixed/Clear
3:0	Reserved	0	R
4	Capabilities Enable. Set to one to indicate Capabilities are enabled.	VS	F
5	Set to one to indicate 66MHz-Capable	VS	F
6	Reserved	0	R
7	Set to one to indicate it is Fast Back-to-Back Capable.	VS	F
8	Set when, in Master mode, a Data Parity error is detected, and bit 6 of the PCI command Register is set to one.	0	C
10-9	DEVSEL timing set to medium speed	V S	F
11	Signaled target abort	0	C
12	Received target abort	0	C
13	Received master abort	0	C
14	Reserved	0	R
15	Detected parity error	0	C
VS means Vendor specific.			
NOTE - F = fixed value; C = may be read by the host. The host may clear the bit to zero by writing a one to the bit.			

7.4.1.5 PCI Revision ID

Indicates the IC revision of the ADMA device implementation.

Address Offset   08h

Default Value   xxh (Vendor Specific)

Attribute   Read Only

Size   8 bits

7.4.1.6 PCI Class Code

Address Offset   09h

Default Value   000105h (see Table 14)

Attribute   Read Only

Size   24 bits

Table 14 – ADMA PCI Class Code

Bits	Description	Value
15-8	Programming Interface Code	20h – Single Stepping 30h – Continuous Operation
23-16	Sub-class Code	05h – ATA
31-24	Base-Class Code	01h – Mass Storage

7.4.1.7 PCI Cache Line Size

In Master mode, the Cache Line Size Register in bytes is used to determine the PCI command appropriate for the burst. The commands implemented are Memory Read, Memory Read Line, Memory Read Multiple, Memory Write, and Memory Write Invalidate. Set by the host BIOS or Operating System.

Address Offset   0Ch

Default Value   00h

Attribute   Read/Write

Size   8 bits

7.4.1.8 PCI Latency Timer

The host writes a value (in PCI clocks) that is decremented by the ADMA during a master mode transfer. If the host deasserts the PCI GNTn signal before this value expires, the ADMA may continue until the latency timer expires or there is no more data to transfer. If the data transfer is complete before GNTn is de-asserted the ADMA terminates the bus master transfer. The host BIOS or Operating System sets this register.

Address Offset   0Dh

Default Value   40h

Attribute   Read/Write

Size   8 bits

7.4.1.9 PCI Header Type

Indicates that the ADMA is a single-function device.

Address Offset   0Eh

Default Value   00h

Attribute   Read Only

Size   8 bits

7.4.1.10 PCI Base Address Registers (BAR)

Base Address Registers 0-3 have bit zero hard-wired to one to indicate I/O space, and bits 16-31 hard-wired to zero. Full address decoding shall be implemented. BARs 4-5 have bit zero hard-wired to zero to indicate host memory address space.

7.4.1.10.1 PCI Base Address 0

This is the base address for the command block registers for ATA Channel X.

Address Offset   10h

Default Value   00000001h

Attribute   Bits 31-16 Read Only, bits 15-3 Read/Write, bits 2-0 Read Only.

Size   32 bits

7.4.1.10.2 PCI Base Address 1

This is the base address for the Control Registers for ATA Channel X. Note that, because of the Dword alignment of PCI, the device Control and Alternate Status Registers are at offset 06h from this base.

Address Offset   14h

Default Value   00000001h

Attribute   Bits 31-16 Read Only, bits 15-3 Read/Write, bits 2-0 Read Only.

Size   32 bits

7.4.1.10.3 PCI Base Address 2

This is the base address for the command block registers for ATA Channel Y. If the device only supports one channel this base address is read only and cleared to zero. Note that, because of the Dword alignment of PCI, the device Control and Alternate Status Registers are at offset 06h from this base.

Address Offset   18h

Default Value   00000001h

Attribute   Bits 31-16 Read Only; bits15-3 Read/Write, bits 2-0 Read Only.

Size   32 bits

7.4.1.10.4 PCI Base Address 3

This is the base address for the Control Registers for ATA Channel Y. If the device only supports one channel this base address is read only and cleared to zero.

Address Offset   1Ch

Default Value   00000001h

Attribute   Bits 31-16 Read Only, bits 15-3 Read/Write, bits 2-0 Read Only.

Size   32 bits

7.4.1.10.5 PCI Base Address 4 and 5

This is the base address for the 64-bit Memory Mapped ATA Channel and ADMA registers.

Address Offset   20h

Default Value   0000000000000004h

Attribute   Bits 63-10 Read/Write; bits 9-0 Read Only. If size is 32 bits then bits 32-63 are Read Only.

Size   32 or 64 bits

7.4.1.11 PCI Subsystem Vendor ID

The PCI subsystem vendor ID indicates the vendor of the adapter.

Address Offset   2Ch

Default Value   xxxxh (PCI Vendor Unique ID)

Attribute   Read Only

Size   16 bits

7.4.1.12 PCI Subsystem ID

The PCI subsystem ID indicates the adapter implementation.

Address Offset   2Eh

Default Value   xxxxh (Vendor Specific)

Attribute   Read Only

Size   16 bits

7.4.1.13 PCI Expansion ROM Base Address

Address Offset   30h

Default Value   If ROM is present this BAR shall contain the size of the ROM in bytes, if no ROM is present this BAR shall contain 00000000h.

Attribute   Bits 31-17 Read/Write; bits 16-01 Read Only; bit zero Read/Write.

Size   32 bits

7.4.1.14 PCI Capability Pointer

The PCI Capability Pointer points to a linked list of capabilities (i.e., the Power Management Registers).

Address Offset   34h

Default Value   50h

Attribute   Read Only

Size   8 bits

7.4.1.15 PCI Interrupt Line

NOTE -   the host BIOS loads the system interrupt (IRQ) allocated to this device. This Register is not used by the ADMA. It may be used by the system BIOS and host software as a location to store the IRQ being used.

Address Offset   3Ch

Default Value   00h

Attribute   Read/Write

Size   8 bits

7.4.1.16 PCI Interrupt Pin

The PCI interrupt pin defaults to 01h, indicating that the PCI interrupt signal is used.

Address Offset   3Dh

Default Value   01h

Attribute   Read Only

Size   8 bits

7.4.1.17 PCI Minimum Grant

The PCI minimum grant is the minimum burst period in units of 250 ns required by the ADMA.

Address Offset   3Eh

Default Value   xxh

Attribute   Read Only

Size   8 bits

7.4.1.18 PCI Maximum Latency

The default value of 00h indicates that the ADMA has no particular requirement for Maximum Latency.

Address Offset   3Fh

Default Value   00h

Attribute   Read Only

Size   8 bits

7.4.1.19 Power Management Registers (optional)

Address Offset   50h

Default Value   see Table 15

Attribute   see Table 15

Size   8 Bytes

Table 15 – ADMA Power Management Registers

Pointer to Next Capability (Read Only)								Capability ID (Read Only)								0h
XXh								01h								
Power Management Capability (Read Only)																2h
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
0	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	
Power Management Control/Status																4h
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
0	0	0	0	0	0	0	R/W	0	0	0	0	0	0	R/W	R/W	
Data Register (Read Only)								Bridge Support Extension (Read Only)								56h
00h								00h								
R/W means that the bit is both read and write capable.																

7.4.1.19.1 Capability ID

The capability ID indicates that the ADMA supports the {PCI PMS}.

7.4.1.19.2 Pointer to Next Capability

The next capability points to the next capability; 00h indicates the end of the linked list of capabilities.

7.4.1.19.3 Power Management Capability

Table 16 – ADMA Power Management Capability Register

Bit	Description
15-14	Reserved.
13	Set to one indicates that the ADMA may assert PME# from the D2 state if the signal UINTRQ (unsolicited interrupt) is asserted on either channel.
12-11	Reserved.
10	Set to one indicates that the ADMA supports the D2 (Standby) state.
9-4	Reserved.
3	Set to one indicates that the ADMA requires a PCI clock to assert PME#.
2-0	Set to 010b indicates that the ADMA complies with version 1.1 of the {PCI PMS}.

7.4.1.19.4 Power Management Control/Status

Table 17 – ADMA Power Management Control/Status Register

Bit	Description
15	Indicates whether PME# can be asserted from power state D3-Cold. Fixed to zero indicating that the ADMA does not support PME# assertion from the D3-Cold state.
14-9	Reserved.
8	Controls the enable and disable of PME#. When this bit is set to one, PME# is enabled. When this bit is cleared to zero, PME# is disabled.
7-2	Reserved.
1-0	Power Management State Control Bits.

7.4.1.19.5 Power Management State Control Bits:

The power management states D0, D2 and D3 are defined in the PCI Power Management Specification.

Table 18 – ADMA Power Management State Control bits.

Bit 1(b1)	Bit 0 (b0)	State
0	0	Active (D0)
0	1	Invalid
1	0	Standby (D2)
1	1	Sleep (D3)

7.4.1.20 Power Management State Transitions

Bits b0 and b1 are power management state control bits defined in Table 18.

D0 : Active   **D2 : Standby**   **D3-Hot : Sleep**

Power On, Reset

:D0

b1:b0
00
01

b1:b0
01
10

b1:b0
01
10
11

D0:D0

D2:D2

D3:D3

b1:b0
10

b1:b0
11

D0:D2

D2:D3

b1:b0
00

D2:D0

b1:b0
11

D0:D3

b1:b0
00

D3:D0
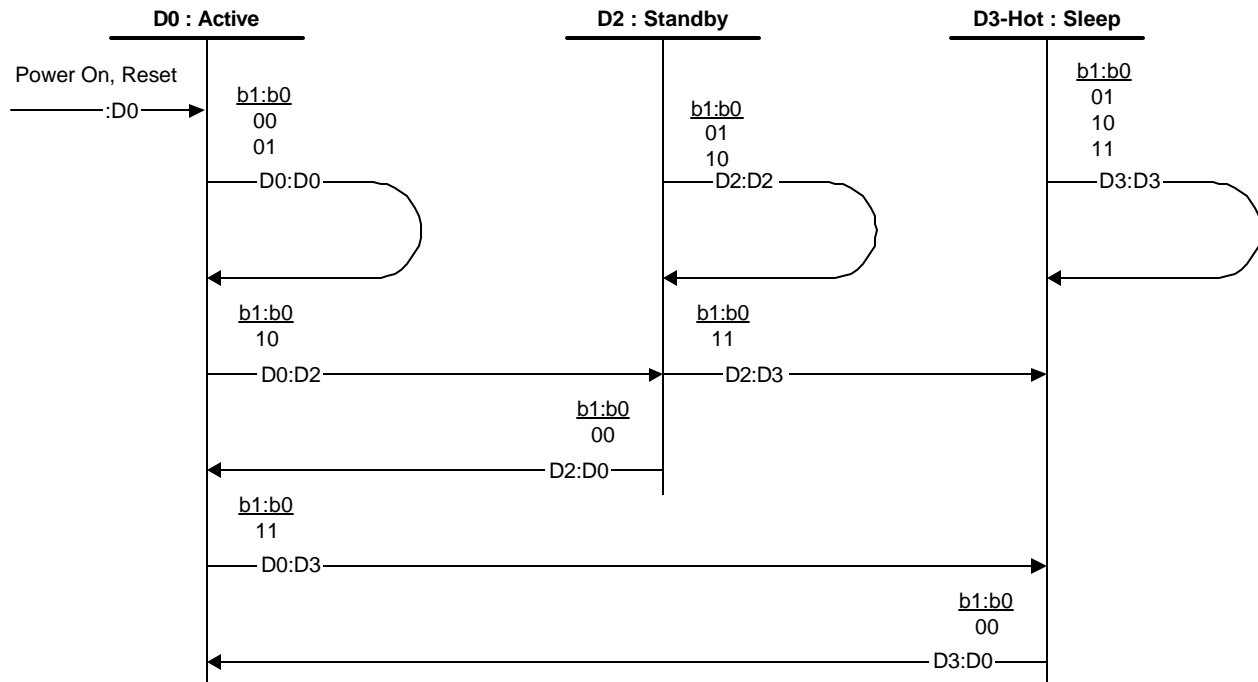
Figure 3 – Power Management State Transitions

7.4.1.20.1 D0:Active State

In this state the PCI bus is able to process all requests and the devices attached to the PCI bus are fully powered.

Transition Initialization:D0: Active is the default power management state for the PCI bus

Transition D0:D2: When the host sets the ADMA Power Management Control bits to 01b, the ADMA controller shall enter the standby state. See 7.4.1.20.2 for more information on the Standby State.

Transition D0:D3: When the host sets the ADMA Power Management Control bits to 11b, the ADMA controller shall enter the D3-Hot: Sleep state. See 7.4.1.20.3 for more information on the D3-Hot: Sleep state.

Transition D0:D0: When the host sets the ADMA Power Management Control bits to 00b, the ADMA controller shall remain in the active state.

7.4.1.20.2 D2: Standby State

When the host sets the ADMA Power Management Control bits to 01b the ADMA controller shall enter the Standby state. In this state the ADMA controller reduces power consumption but can re-enter the active state with no noticeable delay.

Transition D2:D2: When the host sets the ADMA Power Management Control bits to 01b, the ADMA controller shall remain in the standby state.

Transition D2:D3: When the host sets the ADMA Power Management Control bits to 11b, the ADMA controller shall enter the D3-Hot: Sleep state. See 7.4.1.20.3 for more information on the D3-Hot: Sleep state.

Transition D2:D0: When the host sets the ADMA Power Management Control bits to 00b, the ADMA controller shall enter the active state. See 7.4.1.20.1 for more information on the D0: Active State.

7.4.1.20.3 D3-Hot: Sleep

In this State the ADMA controller is powered down although other devices may remain powered. The ADMA controller may be removed from the PCI bus in a Hot-Plug operation once the D3 state has been set.

Transition D3:D3: When the host sets the ADMA Power Management Control bits to 11b or 10b the ADMA controller shall remain in the D3-Hot: Sleep state. Once the D3-Hot: Sleep state has been achieved, the only valid transition out is to enter the active state.

Transition D3:D0: When the host sets the ADMA Power Management Control bits to 00b, the ADMA controller shall enter the active state. See 7.4.1.20.1 for more information on the D0: Active State.

7.5 ADMA Registers

The ADMA register map uses 1024 bytes of host memory space. This includes the ATA Command, Control and Data port Registers. All registers may be accessed as byte, word, or Dword entities. These registers are addressed via the base address in BAR 4 (Bytes 20h-27h) in the PCI configuration header registers. The description of the ADMA registers follows.

Table 19 – ADMA Memory Mapped Registers

Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0	Offset
Reserved		Channel X PIO Data		00h
Reserved			Ch. X Err/Features	04h
Reserved			Ch. X Sector Cnt	08h
Reserved			Ch. X LBA Low	0Ch
Reserved			Ch. X LBA Mid	10h
Reserved			Ch. X LBA High	14h
Reserved			Ch. X Dev. Head	18h
Reserved			Ch. X Stat/Cmd	1Ch
Reserved			ADMA Standard	20h
Reserved				21h-37h
Reserved			Ch. X Alt Stat/Ctrl	38h
Reserved				3Ch
Reserved		Channel Y PIO Data		40h
Reserved			Ch. Y Err/Features	44h
Reserved			Ch. Y Sector Cnt	48h
Reserved			Ch. Y LBA Low	4Ch
Reserved			Ch. Y LBA Mid	50h
Reserved			Ch. Y LBA High	54h
Reserved			Ch. Y Dev. Head	58h
Reserved			Ch. Y Stat/Cmd	5Ch
Reserved				60h-77h
Reserved			Ch. Y Alt Stat/Ctrl	78h
Reserved				7Ch
Reserved	Ch. X ADMA Stat	Chan. X ADMA Control		80h
Reserved		Chan. X ADMA CPB Search Count		84h
Chan. X ADMA Current CPB Address				88h
Chan. X ADMA Next CPB Address				8Ch
Chan. X CPB Lookup Table Address				90h
Chan. X ADMA FIFO Output Threshold		Chan. X ADMA FIFO Input Threshold		94h
Reserved				98h-9Fh
Reserved	Ch. Y ADMA Stat	Chan. Y ADMA Control		A0h
Reserved		Chan. Y ADMA CPB Search Count		A4h
Chan. Y ADMA Current CPB Address				A8h
Chan. Y ADMA Next CPB Address				ACh
Chan. Y CPB Lookup Table Address				B0h
Chan. Y ADMA FIFO Output Threshold		Chan. Y ADMA FIFO Input Threshold		B4h
Reserved				B8h- BFh
Vendor Specific				C0h
Vendor Specific				C4h
Reserved				C8h-D3h
BIOS Message Pointer				D4h
Reserved				D8h
Host Software Message Pointer				DCh
Vendor Specific				E0h-17Fh
Reserved				180h-3FFh

7.5.1 ATA Channel X Command and Control Registers

Reading or writing these registers shall cause a corresponding read or write of the ATA command and control block registers of the ATA devices connected to that channel. The registers may be written or read as 32bit values but only the bits indicated in Table 19 shall either be driven on or read from the ATA channel. All other bits written shall be ignored and shall be read as zero.

7.5.2 ADMA Identify Registers

This register space is used to provide version and other information regarding the capabilities of the ADMA device. Software may determine the type and capability of the device by combining the PCI class code (see 6.6.1) and the information in these registers.

7.5.2.1 ADMA Standard Register

This register indicates which version of the standard the device conforms to.

Address offset   Base + 20h

Default value   00h (Indicates the device conforms to this standard).

Attributes   Read Only

Size   8bits

7.5.2.2 Reserved Identification Registers

Address offset   Base +21h

Default value   cleared to zero.

Size   17Bytes

7.5.3 ADMA Control Register (ADMCTL)

Address offset, Channel X: Base + 80h

Address offset, Channel Y: Base + A0h (If the device does not support channel Y this register shall be read only and cleared to zero).

Default value   100h

Attribute   Read/Write

Size   16 bits

Table 20 – ADMA Control Register

Bit	Name	Reset	Description
15-9		0	Reserved.
8	aIEN	0	PCI channel interrupt disable bit. When cleared to zero, interrupts generated by the channel when in the ATA Register Mode are propagated through to the PCI bus. When set to one, interrupts generated by the channel are not propagated to the PCI bus
7	aGO	0	ADMA GO Bit. When set to one, the ADMA can run. When cleared to zero, the channel operates only in ATA Register Mode. The host writes a one to this bit each time that a CPB has been updated, to notify the ADMA that there is another CPB to service. The ADMA shall not clear this bit. NOTE - When this bit is cleared to zero by the host, the ADMA immediately ceases all operations and goes to ATA Register Mode; the state of the current CPB is indeterminate.
6	aPSE	0	ADMA PAUSE Bit. When set to one, the ADMA does not follow the CPB chain nor access the CPB Lookup Table. If set to one while a CPB is being processed, the ADMA completes the CPB and then PAUSES. The host shall pause operations before modifying the CPB chain pointers by the use of aPSE and aPSD
5	aRSTADM	0	RESET ADMA Channel to the IDLE state. Set to one by the host to indicate a reset is required. Cleared by the host after 1µs to allow the ADMA to come out of the IDLE state.
4			Reserved.
3	aAUTEN	0	ADMA AUTO-POLL ENABLE Bit. When set to one, the ADMA and there is no other activity on the channel the ADMA repeatedly alternates selection of each device on the channel when waiting for interrupts,.This enables a device to assert a Service Interrupt in an overlapped or queued situation.
2	aRSTA	0	ATA HARD RESET Bit. When set to one, the ATA reset signal is asserted. For the host to reset the ATA channel, the host shall set this bit to one, wait for the minimum reset time defined in the {ATA Standard}, and then clear this bit to zero.
1-0	aPIOMD	00	DEFAULT PIO MODE. Used in ATA Register Mode to define the ATA PIO timing. 00 = Mode 1, 01 = Mode 2, 10 = Mode 3, 11 = Mode 4. ATA mode zero is not supported. The value in this register shall be the highest PIO mode supported by the slowest device on the channel. The mode selected is used for all accesses to the ATA command and control block registers, in ATA Register and ADMA Mode. When in ATA Register Mode, the PIO mode selected is used for access to the ATA data port (PIO mode).

7.5.4 ADMA Status Register (ADMSTAT)

Address offset, Channel X: Base + 82h

Address offset, Channel Y: Base + A2h (If the device does not support channel this register shall be read only and cleared to zero).

Default value   See Table 21

Size   8 bits

Attribute   Read/Clear

Can be read by the host at any time. Reading the register clears bits 0, 1, and 7 to zero and de-asserts the PCI Interrupt signal.

Table 21 – ADMA Status Register

Bit	Name	Reset	Description
7	aDONE	0	ADMA DONE Bit. When set to one, indicates the ADMA has finished one or more CPBs.
6	aPSD	1	ADMA PAUSED Bit. When set to one, indicates the ADMA has stopped as a result of aPSE being set. The current transfer has been completed.
5	aSTPD	1	ADMA STOPPED Bit. When set to one, indicates the ADMA has stopped as a result of aGO being cleared, an error occurring, or no more valid CPBs to be processed. See **Error! Reference source not found.** for details of the transitions that result in aSTPD being set to one.
4	aUIRQ	X	ATA UNSOLICITED IRQ Bit. When set to one, indicates the ATA unsolicited interrupt line is active.
3	aLGCY	1	ADMA LEGACY Bit. When set to one, indicates that the ADMA is in ATA Register Mode.
2		0	Reserved.
1	aCPBERR	0	ADMA CPB Error Bit. When set to one, indicates that at least one of the CPB-error response flags in the CPB has been set to one except in the case of cPSEXC (Table 22) and pIGEX (Table 24) set to one.
0	aPERR	0	PCI ERROR Bit. When set to one, indicates that a PCI error has occurred.

7.5.5 ADMA CPB Search Count Register (CCNT)

Address offset, Channel X:	Base + 84h
Address offset, Channel Y:	Base + A4h (If the device does not support channel this register shall be read only and cleared to zero).
Default value	0000h
Attribute	Read/Write
Size	16 bits

The purpose of this register is to limit the number of CPB's scanned by the ADMA. This prevents the ADMA from requiring the PCI bus when there are no valid CPB's. The ADMA CPB Search Count Register holds a value that is copied into the Internal Down-Counter each time either a write occurs to ADMCTL with the ADMA aGO Bit set or the ADMA reads a CPB with the cVLD Bit set and the cDONE Bit clear. The down-counter decrements each time the ADMA reads a CPB with the cVLD Bit clear, cREL set to one or the cDONE Bit set to one. When the down counter reaches zero, the ADMA sets aDONE and transitions to the IDLE state. The value loaded into this register is normally (but not necessarily) related to the number of CPBs in the chain. This value shall be greater than zero.

7.5.6 ADMA Current CPB Address (CCPB)

The ADMA Current CPB Address Register points to the address of the CPB currently being processed. It is loaded by the ADMA whenever a CPB is read or, in the case of queued operation, the CPB Lookup Table is read.

Address offset, Channel X:	Base + 88h
Address offset, Channel Y:	Base + A8h (If the device does not support channel this register shall be read only and cleared to zero).
Default value	00000000h
Attribute	Read Only
Size	32 bits

7.5.7 ADMA Next CPB Address (NCPB)

Address offset, Channel X: Base + 8Ch

Address offset, Channel Y: Base + ACh (If the device does not support channel this register shall be read only and cleared to zero).

Default value   00000000h

Attribute   Read/Write

Size   32 bits

The ADMA Next CPB Address Register is initialized by the host to point to the first CPB in a chain that the host has constructed in host memory. The address shall be a physical address. This register is updated by the ADMA each time it reads a CPB, except when retrieving a CPB from the Lookup Table. The host shall not write to this register unless the ADMA is in the Legacy Idle or the Paused State (see 7.7). The CPB pointed to by this register may not physically follow the CPB pointed to by the CCPB pointer.

Note that the host software shall ensure that the content of each CPB within the chain shall be physically contiguous and locked in host memory and that all chain pointers shall be physical addresses.

7.5.8 CPB Look Up Table Address Register (CPBLAR)

Address offset, Channel X: Base + 90h

Address offset, Channel Y: Base + B0h (If the device does not support channel this register shall be read only and cleared to zero).

Default value   00000000h

Attribute   Read/Write

Size   32 bits

The CPB Lookup Address Register contains the 32-bit address of the base of the CPB Lookup Table. The host software initializes the contents of this register before entering ADMA Mode. The ADMA uses this address as a base to which it adds the offset calculated from the device, and tag resulting from a Service command, the offset calculation is defined in 7.6.2. The resulting calculated address points to a location in host memory that, in turn, points to the CPB associated with the command requiring service.

The CPB Lookup Table Address Register is only used in overlapped or queued operations. It is initialized by the host to point to the base of a Lookup Table that it is constructed in host memory. The CPB Lookup Table shall be physically contiguous and locked in host memory. The address contained in this register shall be a physical address. This register is used by the ADMA to construct the address of the applicable CPB when a service interrupt is received. Each entry in the table is a Qword of which bits 0-31 are significant, bits 32-63 shall be cleared to zero by the host. Bits 0-31holds the physical address of the CPB.

7.5.9 ADMA FIFO Input Threshold Register (FITR) (optional)

Address offset, Channel X: Base + 94h

Address offset, Channel Y: Base + B4h (If the device does not support channel this register shall be read only and cleared to zero).

Default value   0000h

Attribute   Read/Write

Size   16 bits

The ADMA FIFO Input Threshold is initialized by the host to request the desired PCI burst length. It represents the number of Qwords in the Input FIFO to the ADMA from the PCI bus before the FIFO Input Threshold (FIT) Flag sets and a PCI burst is initiated to write the data to host memory. The value loaded into this register may be related to the value in the CacheLine Size Register (offset 0Ch) in the PCI configuration Registers. The value of FITR shall be greater than zero and less than the FIFO size.

7.5.10 ADMA FIFO Output Threshold Register (FOTR) (optional)

Address offset, Channel X: Base + 96h

Address offset, Channel Y: Base + B6h (If the device does not support channel this register shall be read only and cleared to zero).

Default value   0000h

Attribute   Read/Write

Size   16 bits

The ADMA FIFO Output Threshold is initialized by the host to request the desired PCI burst length. It represents the space, in Qwords, in the Output FIFO from the ADMA to the PCI bus before the FIFO Output Threshold (FOT) Flag sets and a PCI burst is initiated to read data from host memory. The value loaded into this register may be related to the value in the Cache Line Size Register (offset 0Ch) in the PCI configuration Registers. The value of FOTR shall be greater than zero and less than the FIFO size.

7.5.11 BIOS Message Pointer

Address Offset   D4h

Default Value   00000000h

Attribute   Read/Write

Size   32 bits

This register may be used by the BIOS to point to its message area. The ADMA takes no action on this register other than clearing it to zero on reset. This allows a BIOS to provide information to the host software.

7.5.12 Driver Message Pointer

Address Offset   DCh

Default Value   00000000h

Attribute   Read/Write

Size   32 bits

The host software may use this register to point to its message area. The ADMA takes no action on this register other than clearing it zero on reset. This allows host software to provide information to a BIOS.

7.6 Auto DMA Mode Data Structures

In ADMA Mode the ADMA engine reads a command held in a command chain from host memory. The command part of a data structure termed a Command Parameter Block (CPB). A chain of CPBs is created in host memory, with each CPB pointing to the next CPB. The CPB pointer shall be a physical address. The ADMA maintains a Dword register that points to the next CPB of the chain. Within each CPB, there is a pointer to the chain of ADMA Physical Region Descriptors (APRDs). The APRD is a structure that defines the memory locations where the data is to be written to or read from, see 7.6.1.8. The APRD pointer shall be a physical address.

7.6.1 Command Parameter Block

The CPB is a block of parameters and commands for the ADMA and, indirectly, for the ATA Channel. Each CPB shall all be physically contiguous, locked in host memory, and possibly Qword-aligned in physical address space.

NOTE – Some host controllers require Qword alignment of CPB addresses in physical address space. However, this requirement is optional in this standard.

CPB data, described in Table 22, is written by the host; only the Response Flags are modified by the ADMA.

Table 22 – CPB Structure

Qword	Byte	Bits	Name	Init	Description
0	Response Flags	0	0	cDONE	1	The host shall clear this bit to zero to give the ADMA control of the CPB. The ADMA shall set this bit to one to give the host control of the CPB. See 7.6.3.
			1	cREL	0	The ADMA shall set cREL to one if the device indicates that the command associated with the CPB is to be released. Section 7.6.3.
			2	cIGNRD	0	The ADMA shall set cIGNRD to one if, on the first access of the CPB, cVLD, cREL and cDONE are cleared to zero. See 7.6.3.
			3	cATERR	0	The ADMA shall set cATERR to one if ATA ERR is set to one during execution of the command.
			4	cSPNT	0	The ADMA shall set cSPNT to one if it detects a spurious interrupt on the ATA INTRQ signal during execution of a command.
			5	cPSDEF	0	The ADMA shall set cPSDEF to one if the APRD data transfer lengths are insufficient to complete the command. This means that the DMA engine in the ADMA may still be running.
			6	cPSEXC	0	The ADMA shall set cPSEXC to one if the APRD data transfer length is in excess of that required to complete the command. . This means that the DMA engine in the ADMA may have stopped while the device is still attempting to transfer data.
			7	cCPBERR	0	The ADMA shall set cCPBERR to one if it determines that the CPB is inconsistent.
		1	7-0		0	Reserved.
	Control Flags	2	0	cVLD	0	cVLD is used in combination with cDONE and cREL to control the processing of the CPB by the ADMA engine. When cDONE is set to one, the CPB shall not be processed. See 7.6.3.
			1	cQUE	0	cQUE shall be cleared to zero by the host for non-queued or non-overlapped commands. cQUE shall be set to one for queued/overlapped commands.
			2	cDAT	0	cDAT shall be set to one by the host to indicate that the APRD chain contains valid information. The address in cPRD shall be valid.
			3	cIEN	0	cIEN is cleared to zero by the host to disable the command complete interrupt; set to one by the host to allow the Command Complete interrupt.
			7-4		0	Reserved.
	ATA Length	3	7-0	cLEN	0	The length in Qwords of the ATA Register Field Area of the CPB (the total CPB length = 2 + cLEN Qwords).
		4-7	31-0	cNCPB	x	Host memory Address of the next CPB. Shall be Qword aligned in some implementations.
1	APRD	0-3	31-0	cPRD		Host memory Address of the first APRD for this CPB. Shall be Qword aligned in some implementations.
		4-7			0	Reserved.
2	CMD	0-1	15-0	ATAR0	0's	ATA Register Field – see 7.6.1.6
	CMD	2-3	31-16	ATAR1	0's	ATA Register Field – see 7.6.1.6
	CMD	4-5	47-32	ATAR2	0's	ATA Register Field – see 7.6.1.6
	CMD	6-7	63-48	ATAR3	0's	ATA Register Field – see 7.6.1.6
…	CMD	…	…	…	…	…
n	CMD	0-1	15-0	ATAR0	0's	ATA Register Field – see 7.6.1.6
	CMD	2-3	31-16	ATAR1	0's	ATA Register Field – see 7.6.1.6
	CMD	4-5	47-32	ATAR2	0's	ATA Register Field – see 7.6.1.6
	CMD	6-7	63-48	ATAR3	0's	ATA Register Field – see 7.6.1.6

7.6.1.1 Response Flags (Byte 0)

These flags are written by the ADMA during the processing of the CPB. Bit zero (cDONE) shall be set to one by the host when preparing the CPB. The host shall clear the byte to zero to indicate to the ADMA that the CPB is

valid and ready to be processed. The Response Flags are in a byte by themselves, so that the ADMA does not have to do a read/modify/write operation.

7.6.1.1.1 cDONE – ATA Command Complete Flag (Bit 0)

This flag shall be set to one by the ADMA hardware when it has completed processing the command in this CPB entry. It is used by the ADMA to prevent processing the CPB again on subsequent passes around the CPB chain. When set, the host has control of the CPB.

The host sets cDONE to one when initializing a CPB in the CPB chain. The host first sets cVLD to one and then clears cDONE to zero to indicate to the ADMA that the CPB contains valid command information. Thereafter, with the exception of cVLD, the host shall not change anything in the CPB until cDONE is set to one by the ADMA. The host can then write new command information to the CPB, set cVLD to one and, finally, clear cDONE to zero. See 7.6.3.

7.6.1.1.2 cREL – ATA Release Interrupt Flag (Bit 1)

This flag shall be set to one by the ADMA hardware when the ATA REL Bit is set to one by the device after a queued command has been written to the device. When the ADMA has set this flag it proceeds to the next Valid-Waiting CPB, unless the ATA SERV Bit is set to one. In this latter case, the ADMA then issues a service command to the device to process a previously queued command. See 7.6.3.

7.6.1.1.3 cIGNRD – CPB Ignored (Bit 2)

If the ADMA hardware reads a CPB with both cDONE and cVLD cleared to zero, it sets both cDONE and cIGNRD to one, sets aDONE, and asserts the PCI interrupt. The CPB is ignored and the next CPB in the chain is processed. cVLD is assumed to be set to one if the CPB is being accessed due to a Service Interrupt. See 7.6.3. The PCI interrupt will not be asserted if the CPB is configured to mask interrupts.

7.6.1.1.4 cATERR – ATA Command Error Flag (Bit 3)

This flag shall be set to one by the ADMA hardware if the ATA ERR (ATAPI CHK) bit set in the ATA Status (or Alt Status) Register during the command. When the ADMA sets this bit, it sets aCPBERR, asserts the PCI Interrupt signal, and transitions to ATA Register Mode. The host is responsible for error recovery.

7.6.1.1.5 cSPNT – ATA Spurious Interrupt Error Flag (Bit 4)

This flag shall be set to one by the ADMA hardware if the ATA INTRQ signal is asserted unexpectedly during execution of a command. When the ADMA sets this bit, it sets aCPBERR, asserts the PCI Interrupt signal regardless of the state of cIEN, and transitions to ATA Register Mode. The host is responsible for error recovery.

7.6.1.1.6 cPSDEF – APRD Deficiency Length Error Flag (Bit 5)

This flag shall be set to one by the ADMA hardware if the total transfer length in the APRD chain is insufficient to complete the ATA transfer. In this situation the ATA device might be hung or data might be lost. When the ADMA sets this bit, it sets aCPBERR, asserts the PCI Interrupt signal (irrespective of the state of cIEN), and transitions to ATA Register Mode. The host is responsible for error recovery.

7.6.1.1.7 cPSEXC – APRD Excess Length Error Flag (Bit 6)

This flag shall be set to one by the ADMA hardware if the transfer is complete before the APRD length expires. In this case, the device will have completed the command, with or without errors. When the ADMA sets this bit, it sets aCPBERR, may assert the PCI Interrupt signal, and may transition to ATA Register Mode depending on the state of pIGEX (see 7.6.1.7). The host is responsible for error recovery.

7.6.1.1.8 cCPBERR – ATA Command Error Flag (Bit 7)

This flag shall be set to one by the ADMA hardware if it detects an inconsistency in the CPB. When the ADMA sets this bit, it sets aCPBERR, asserts the PCI Interrupt signal (irrespective of the state of cIEN), and transitions to ATA Register Mode. The host is responsible for error recovery.

7.6.1.2 Control Flags (Byte 2)

These flags control the detailed operation of the ADMA sequencer. They remove the need for the ADMA to recognize the ATA command set. Thus, if new commands are defined, the ADMA can still function.

7.6.1.2.1 cVLD – CPB Valid (Bit 0)

The host shall set cVLD to one to indicate that the CPB will be processed when cDONE is cleared to zero. The host shall not set cDONE to one (unless it is initializing the CPB chain). If the host determines that a CPB need no longer be processed, it may clear cVLD to zero. Note that this does not necessarily guarantee that the CPB will be ignored. If the CPB is accessed by the ADMA after this bit is set, the ADMA ignores the CPB and indicates such by setting cIGNRD to one. If the CPB is in the Released State, the ADMA ignores cVLD when accessing the CPB in response to an ATA service request interrupt. To check that a command was "ignored" after cVLD has been cleared, the host shall check the state of cIGNRD after cDONE has been set to one by the ADMA. See 7.6.3 for more information.

7.6.1.2.2 cQUE – Overlap/Queue Flag (Bit 1)

This flag shall be set to one to indicate that the command set contains an overlapped/queued command. This flag cleared to zero indicates that there is no overlapped/queued command. If this flag is set, the ADMA inspects the ATA SERV Bit and the ATA REL Bit on the assertion of the ATA INTRQ signal.

7.6.1.2.3 cDAT – APRD Valid Flag (Bit 2)

The host shall set this flag to one to indicate that cPRD is valid. The APRD chain may contain Directed Interrupt Information, ATAPI Packet data pointers, data transfer pointers, or any combination of these.

7.6.1.2.4 cIEN – PCI Interrupt Enable Flag (Bit 3)

The host shall clear this flag to zero to prevent the ADMA from generating the PCI Interrupt signal when the command is complete. The host shall set this flag to one to allow the PCI Interrupt signal. Clearing this flag will not prevent the PCI Interrupt signal from being asserted in the event of an error.

7.6.1.3 cLEN – ATA Length (Byte 3)

This Byte contains the number of Qwords that follow the second Qword of the CPB. This enables the ADMA to correctly request the number of Qwords to fetch for any particular CPB.

7.6.1.4 cNCPB - Next CPB Address (Dword 1)

During initialization the host shall construct in memory a chain of CPBs, each of which is Qword-aligned in physical address space. Qword alignment is optional in some implementations. Each CPB shall have in this field the physical address of the next CPB. The host shall write the address of the first CPB into the ADMA Next CPB Address Register before setting the aGO Bit in ADMCTL.

If the host needs to change the chain pointers while the ADMA is running, it shall first pause the ADMA by setting the aPSE Bit in ADMCTL, and checking that the aPSD Bit in ADMSTAT has been set. This prevents the ADMA from using any a pointer that might be invalid.

7.6.1.5 cPRD – APRD Address (Dword 2)

The host at initialization shall construct a APRD chain, as required, so that each CPB has a corresponding APRD chain with its physical starting address in this field.

7.6.1.6 ATARn ATA Register Field

The ATA Register Field is a list of Qwords describing the ATA register writes involved in setting up an ATA command. Each Qword consists of four 16-bit entries. Each entry defines an ATA register write. There may be as many of these Qwords included in a CPB as required (see Table 23.)

Table 23 – ATA Register Field

Control			Address					Data							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	WNB	IGN	CS1-	CS0-	DA2	DA1	DA0	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	WNB	IGN	CS1-	CS0-	DA2	DA1	DA0	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
0	WNB	IGN	CS1-	CS0-	DA2	DA1	DA0	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
END	WNB	IGN	CS1-	CS0-	DA2	DA1	DA0	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

Each 16-bit entry consists of eight bits that define the register content to be written, five bits that define the address of the register to be written, with three bits being used for control purposes.

The three control bits are Ignore (IGN), Wait-Not-Busy (WNB), and End (END). IGN is used to indicate to the ADMA that this entry is to be ignored, and to skip to the next entry. WNB is used to indicate to the ADMA that it shall wait for the device to become not busy before writing the data. The END Bit (bit 63 of the last Qword) indicates that the current entry is the last one to be processed, and shall be the last entry of a set. The ADMA reads the ATA register field entries. When the ADMA detects the END Bit set, it stops reading.

The data (DD0-DD7) and address (DA0-DA2) bits are active high (Asserted = 1). Bits CS0- and CS1- are active low (Asserted = 0).

Each CPB shall contain a maximum of 1 ATA command.

If END is set to one, the host shall not set IGN to one.

7.6.1.7 APRD Chain

Each APRD Chain may contain a variable number of entries (APRDs). The APRD entry shall be physically continuous, locked in host memory, and Qword-aligned in physical address space. Qword alignment is optional in some implementations. The information in the APRD entry is derived by the host, and describes the physical addresses corresponding to the logical buffer address in the original I/O request. There can be several APRDs to describe a transfer buffer because some processors fragment physical memory by the use of paging registers.

In the case of an ATAPI Packet Command, the first APRD is used to describe the packet itself.

In the case of Directed Interrupts, the APRD contains the target Memory or I/O address and the Data to be written to the address.

7.6.1.8 ADMA Physical Region Descriptor

Each APRD, described in Table 24, is two Qwords in size and points to a region of host memory or an I/O address. The ADMA engine reads each APRD in turn, and transfers data to or from the APRD associated memory block or I/O address, until the ATA device interrupts to indicate the end of the transfer.

Table 24 – APRD Data Structure

Qword	Byte	Bits	Name	Description
0	3-0	31-0	pMAD	Physical address of the start of a physically contiguous host memory region. Shall be Qword-aligned in some implementations. If an I/O transfer, the I/O address of the source or destination of the data.
	7-4	31-0	pLEN	If pPKT is cleared to zero, pLEN indicates the length, in Qwords, of the transfer segment. If pPKT is set to one and pDINT is cleared to zero, pLEN indicates the length, in words, of the total data transfer of all the subsequent APRDs (see pPKT). If pPKT is set to one and pDINT is set to one, pLEN contains a 32-bit message (see pDINT).
1	0	0		Reserved.
		1	pIGEX	Ignore Data Excess. Set to one to indicate to the ADMA that data excess occurring in this APRD is not an error. This is primarily used when reading the results from certain ATAPI packet commands that return unknown or odd lengths of data. cPSEXC will be set but no error interrupt will be generated and the ADMA continues execution.
		2	pPKT	Set to one to indicate that pMAD is a pointer to a Packet (pPKLW indicates the length of the packet). pLEN indicates the total length of the transfer found in subsequent APRDs. pDINT shall be cleared to zero when pPKT is set to one.
		3	pDINT	Set to one to indicate that a Directed Interrupt (DINT) is to be performed, if a non-error interrupt event occurs. pMAD is the host memory or I/O address into which to write a 32-bit message contained within pLEN. Note that the PCI interrupt signal is controlled by cIEN only (both Directed Interrupts and the PCI interrupt signal may be enabled, depending on the respective states of pDINT and cIEN).
		4	pORD	Data Transfer method. Set to one for Ultra DMA, cleared to zero for DMA-assisted PIO.
		5	pDIRO	Data Transfer Direction. Shall be set to one for output from the ADMA to the ATA device, cleared to zero for input from the ATA device to the ADMA.
		6	pIOM	Set to one for I/O transfers, cleared to zero for memory transfer.
		7	pEND	In the last APRD of a APRD chain, pEND shall be set to one and pNXT cleared to zero.
	1	3-0	pTMOD	PIO mode or Ultra DMA mode to use, depending on pORD. PIO mode zero is not supported, and PIO modes are decremented by one, meaning PIO mode one is indicated by a zero in this field, PIO mode 2 by a 1, etc. The Ultra DMA modes (0-5) are fully supported (mode 0 = 0…mode 5 = 5).
		6-4	pCRC	If pORD is set to one, this field shall define the burst size that the ADMA will use before terminating and sending a CRC. A value of 000b indicates that all data for this command has been transferred. Values 001b to 111b indicate the burst size in 512-Byte units.
		7		Reserved.
	2	7-0	pPKLW	Packet length in words if pPKT is set to one.
	3	7-0		Reserved.
	7-4	31-0	pNXT	Physical address of the next APRD. In the last APRD of a APRD chain, pNXT is cleared to zero.

If the ATA device attempts to transfer more data than is specified by the APRD chain, the ADMA sets cPSDEF and cCPBERR to one, transitions to ATA Register Mode, and asserts the PCI interrupt signal. The ADMA engine shall ensure that the device completes the command.

If the ATA device attempts to transfer less data than is specified by the APRD chain, when pIGEX is cleared to zero the ADMA sets cPSEXC and cCPBERR to one, transitions to ATA Register Mode, and asserts the PCI Interrupt signal.

If the device has completed the command and the last APRD space is not exhausted and pIGEX is set to one, the ADMA transitions to ATA Register Mode after first setting cPSEXC and aCPBERR to one, and then asserts the PCI Interrupt signal.

7.6.2 CPB LookUp Table

If an overlapped or queued operation is required, the host shall construct a CPB Lookup Table (see Figure 2) and writes the base address of the CPB Lookup table in CPBLAR prior to starting the ADMA. The table shall be physically contiguous, locked in host memory and Qword-aligned in physical address space. Qword alignment is only required in some implementations. Each entry shall be a Qword with the low-order Dword containing the physical address of the CPB, and the high-order Dword cleared to zero. Note that, if the host modifies the CPB chain while the ADMA is in the Paused State (see 7.7), this table shall be updated before restarting the ADMA.

The CPB Lookup Table is used to fetch the original CPB in order to access the APRD chain that controls the transfer of the data. In this case, when an ATA interrupt is received with the ATA SERV Bit set, the ADMA retrieves the ATA TAG field from the device and uses it to construct an address within this table. The address calculation is:

$$\text{Contents of CPBLAR} + (\text{DEV} ? 100h) + (\text{TAG} ? 08h)$$

Where DEV is the ATA DEV Bit, and TAG is the ATA TAG.

7.6.3 CPB States

The CPB can be in one of four States: Not-Valid, Valid-Waiting, Valid-Processing, or Released (see Figure 4). These States are controlled by three bits in the CPB structure: cDONE, cREL, and cVLD.
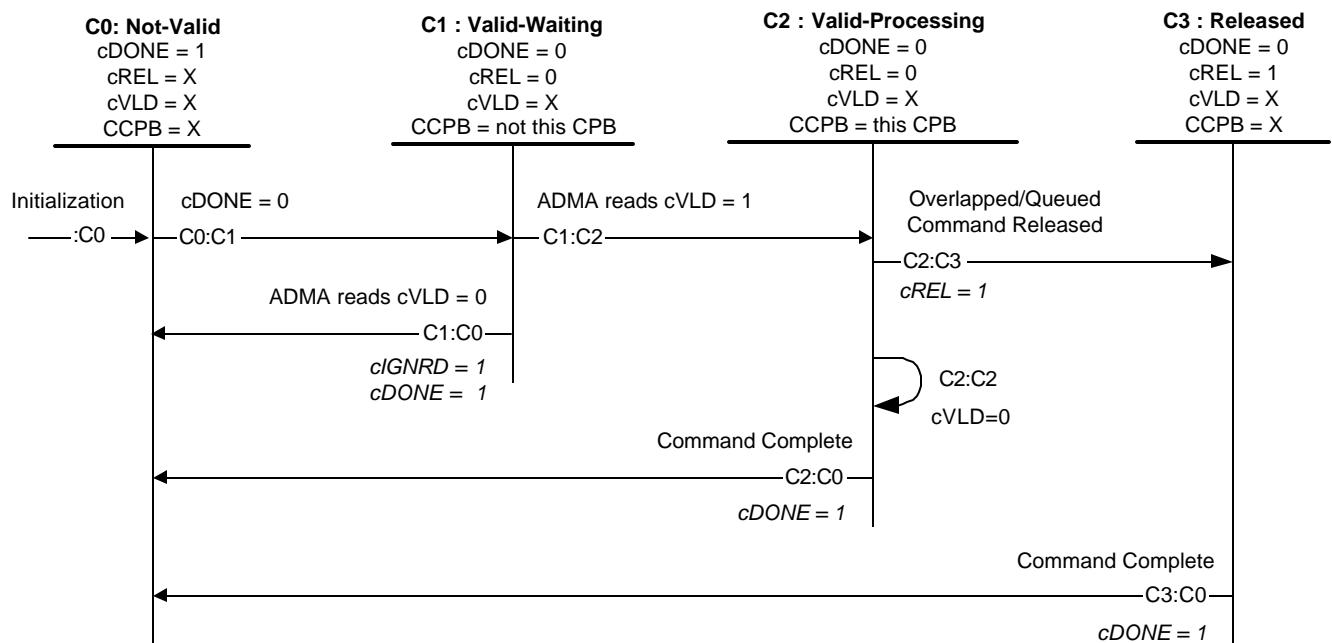

Figure 4 – CPB States

7.6.3.1 C0: Not-Valid State

The host controls a CPB when it is in the Not-Valid State. The ADMA does not initiate execution of a CPB that is in this State.

Transition Initialization:C0: A CPB enters the Not-Valid State when cDONE is initialized to one by the host.

Transition C0:C1: To execute a CPB, the host shall clear the Response Flags Byte to zero in the CPB. This shall transition the CPB to the Valid-Waiting State.

7.6.3.2 C1: Valid-Waiting State

When in this state the CPB is waiting to be processed by the ADMA.

Transition C1:C0: The ADMA detects cVLD equal to zero before it begins execution of the CPB.

Page 43

The host may cause the CPB to be ignored by clearing cVLD to zero while in the Valid-Waiting State.

NOTE - the host is unable to differentiate between the Valid-Waiting and Valid-Processing States. This means that the host clearing cVLD to zero may or may not cause a CPB to be ignored.

Transition C1:C2: When the ADMA accesses a Valid-Waiting CPB, the CPB transitions to the Valid-Processing State if cVLD = 1.

7.6.3.3 C2: Valid-Processing State

In this State the ADMA delivers the ATA command contained within the CPB to the ATA device.

Transition C2:C0: Upon completion of the ATA command(s), the ADMA sets cDONE to one, transitioning the CPB from the Valid-Processing State to the Not-Valid State.

Transition C2:C3: If the ATA device sets the ATA REL Bit to one, cREL is set to one, cDONE is not set, and the CPB transitions to the Released State.

Transition C2:C2: If the host clears cVLD to zero while in the Valid-Processing State, the ADMA engine continues processing the CPB.

7.6.3.4 C3: Released State

The CPB is in this state when the ATA overlapped or queued command contained in a Valid-Processing CPB has been loaded into the ATA device, and the device clears BSY to zero with the ATA REL Bit set to one.

Transition C3:C0: Upon completion of the released ATA command(s), the ADMA sets cDONE to one, transitioning the CPB from the Released State to the Not-Valid State.

7.7 ADMA Operation

7.7.1 Operational Modes and States

The ADMA engine is either in ATA Register or ADMA Mode. The ADMA can be in one of four States: Legacy Idle, ADMA Idle, Run, or Paused. When the ADMA is in the Legacy Idle State, the ADMA engine is in ATA Register Mode. When the ADMA is in any other State, the ADMA engine is in ADMA Mode.

7.7.2 ADMA States

The ADMA engine operates as a state machine (see

). The host software controls the ADMA state via the aGO, and aPSE Bits in ADMCTL. The host software may check the state of the ADMA via the aPSD, aSTPD, and aLGCY Bits in ADMSTAT.

The ADMA is controlled by the values in ADMCTL, CCNTR, ATA ERR, and the ATA Service Interrupt. The ADMA reports its status in ADMSTAT.

indicates the States and the transitions between them.
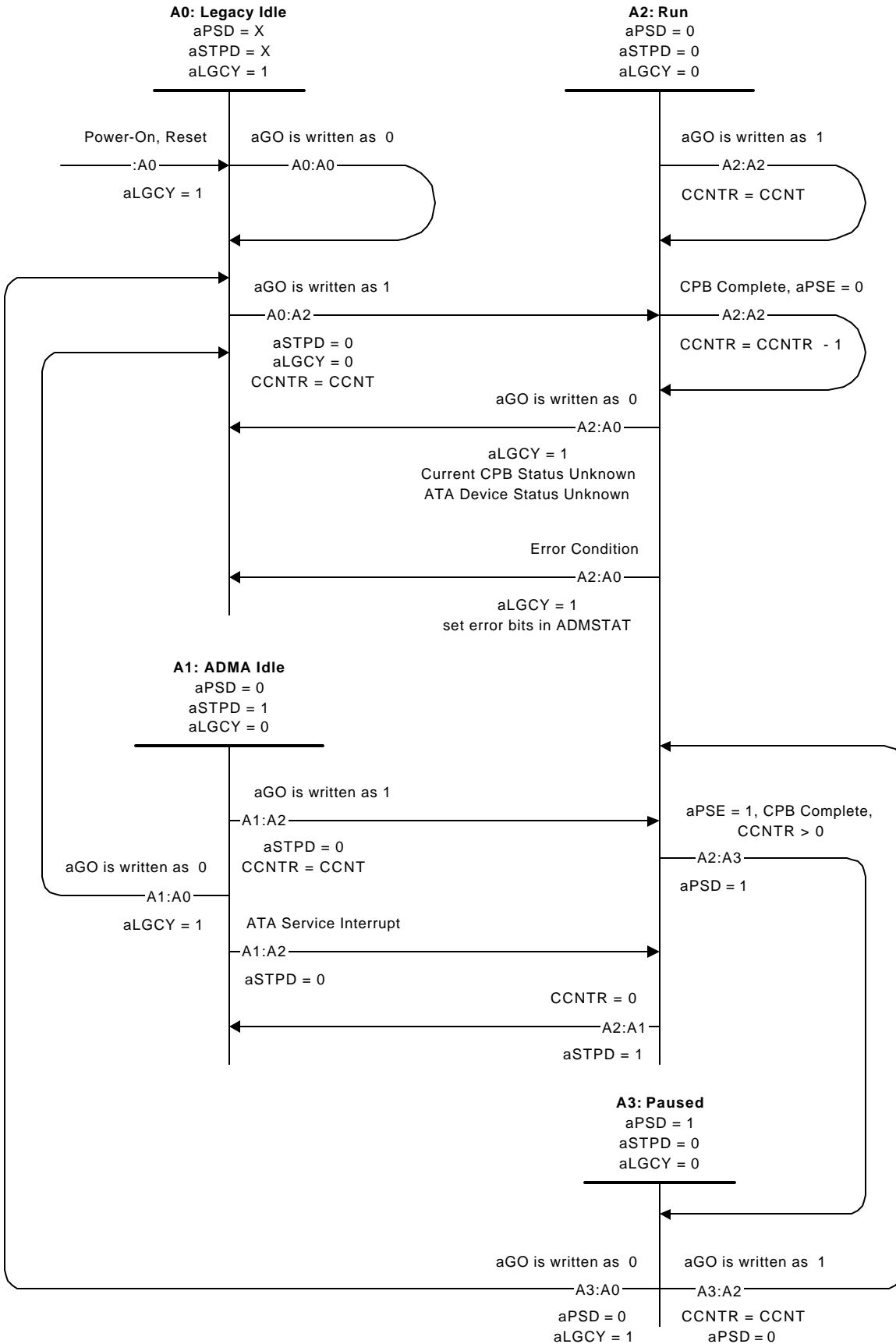
Figure 5 – ADMA State Transitions

7.7.2.1 A0: Legacy Idle State

The Legacy Idle State is the power-on default State. In this State, the ADMA acts as an address decoder for the host. All reads and writes are performed using host I/O or host memory instructions. The only function performed by the ADMA is to control the signal timings of the ATA bus using the ATA core, and to respond to PCI signals. In the Legacy Idle State, all data transfers use the PIO protocols, and ATA bus interrupts are directly mapped onto the PCI Interrupt signal.

Transition A0:A0: Legacy Idle to Legacy Idle

The host's writing aGO as zero leaves the ADMA in the Legacy Idle State.

Transition A0:A2: Legacy Idle to Run

The host's loading a value greater than zero into CCNT and setting aGO to one will transition the ADMA from Legacy Idle to Run. CCNTR will be initialized with the contents of CCNT. The ADMA continues to execute CPBs until the Run State is exited (because aGO is cleared to zero, aPSE is set to one, or CCNTR decrements to zero).

7.7.2.2 A1: ADMA Idle State

In this State, the ADMA takes no actions.

Transition A1:A2: ADMA Idle to Run

When the host writes aGO as one, the ADMA engine transitions from ADMA Idle to Run, and CCNTR is initialized with the contents of CCNT. If aPSE = 1, a single CPB will be executed. If aPSE = zero, the ADMA engine executes CPBs continuously.

If the ADMA receives an ATA Service interrupt the ADMA transitions from ADMA Idle to Run automatically without refreshing the value in CCNTR.

Transition A1:A0: ADMA Idle to Legacy Idle

The host's clearing aGO to zero shall transition the ADMA to the Legacy Idle State.

7.7.2.3 A2: Run State

In this State the ADMA reads and executes CPBs.

Transition A2:A2: Run to Run

When the host writes aGO as one, CCNTR is refreshed with the contents of CCNT. The CPB in process is not affected. CCNTR shall always refreshed when aGO is set.

When the current CPB is completed, CCNTR is decremented.

Transition A2:A0: Run to Legacy Idle

When a transition from the Run to the Legacy Idle State occurs, the host shall take the necessary steps to bring both the ATA device and the ADMA into a known state. This may include resetting the ADMA and/or the device.

When the host clears aGO to zero, the ADMA transitions immediately from the Run State to the Legacy Idle State. This action is not recommended: the status of the current CPB and of the device is unknown.

When an error condition occurs, the ADMA transitions from the Run State to the Legacy Idle State. This transition indicates to the host that an ADMA recognized error has occurred. See 7.7.4, for a complete discussion of ADMA recognized error conditions.

Transition A2:A3: Run to Paused

If aPSE and aGO are set to one when the current CPB is completed, the ADMA transitions to the Paused State.

Transition A2:A1: Run to ADMA Idle

The ADMA transitions to the ADMA Idle State when CCNTR decrements to zero. aGO is unchanged by the ADMA as a result of this transition. If a Valid-Processing CPB is completed at the same time, the ADMA sets aDONE to one, and may assert the PCI Interrupt signal.

7.7.2.4 A3: Paused State

In this State, the ADMA takes no actions.

Transition A3:A0: Paused to Legacy Idle

If the host clears aGO to zero, the ADMA transitions to the Legacy Idle State.

Transition A3:A2: Paused to Run

When the host sets aGO to one and clears aPSE to zero, the ADMA transitions to the Run State and continuously executes CPBs.

When the host sets aGO to one and aPSE to one, the ADMA transitions to the Run State and executes one CPB. See A.7.

In the event that an ATA Service Interrupt has occurred while the ADMA was in the Paused State, the ADMA may process the Service Request(s) before executing any non-Released CPB.

7.7.3 Interrupt Assertion

The ADMA asserts the PCI Interrupt signal when it sets a bit in ADMSTAT. See aIEN in Table 20, and cIEN in Table 22 for exceptions.

NOTE - if an error occurs, CCPB may not point to the CPB with the error.

7.7.4 Error Handling

The ADMA detects the following types of error condition: ATA Error, ATA Spurious Interrupt, CPB Error, APRD deficiency, and APRD excess. Each of these errors are reported in the CPB Response Byte.

The ADMA also detects PCI errors. A PCI Error is reported by aPERR being set to one in ADMSTAT.

In all instances of an error occurring, the ADMA transitions to the Legacy Idle State. When an error has occurred, the ADMA ignores a write of one to aGO until ADMSTAT is read, clearing aCPBERR.

7.7.4.1 ATA Error

When the ADMA detects ATA ERR set to one, it sets cATERR and aCPBERR to one, transitions to the Legacy Idle State, and asserts the PCI Interrupt Signal. (The host software shall assume that the CPB did not complete successfully.) In ATA devices, this indicates that an error has occurred. In ATAPI devices, this might indicate an error or a check condition.

7.7.4.2 ATA Spurious Interrupt

If the ATA INTRQ signal is unexpectedly asserted while the ADMA is in the Run State, the ADMA shall set cSPNT to one, shall set aCPBERR to one, shall transition to the Legacy Idle State, and shall assert the PCI interrupt signal.

For example a spurious interrupt may indicate a faulty ATA channel or a device malfunctioning. Data transfers in progress are stopped.

When an ATA spurious interrupt occurs, the host regains control of the ATA channel by toggling aRSTADM followed by aRSTA in ADMCTL.

7.7.4.3 CPB Error

This error occurs when a service interrupt points to a CPB that is not in the Released State. The ADMA shall set cCPBERR to one, sets aCPBERR to one, transitions to the Legacy Idle State, and asserts the PCI Interrupt signal. This indicates that an ATA TAG, or an ADMA data structure, may have been corrupted.

7.7.4.4 APRD Deficiency

This error occurs when the APRD transfer lengths are insufficient to complete the command. The ADMA sets cPSDEF to one, sets aCPBERR to one, transitions to the Legacy Idle State, and asserts the PCI Interrupt signal. The ADMA engine shall ensure that the device completes the command.

7.7.4.5 APRD Excess

This error occurs when the APRD transfer lengths are in excess of that required to complete the command. The ADMA sets cPSEXC to one irrespective of the state of pIGEX. If pIGEX is cleared to zero in the APRD, the ADMA sets aCPBERR to one in ADMSTAT, transitions to the Legacy Idle State, and asserts the PCI Interrupt signal.

7.7.4.6 PCI Error

The ADMA detects a PCI error whenever bits 8, 12, 13, or 15 of the PCI Status Register are set to one, indicating a severe system problem. See {PCI Spec}. Any transfers across the PCI bus may result in catastrophic failure. The ADMA ceases all ATA operations, sets aPERR to one, transitions to the Legacy Idle State, and asserts the PCI Interrupt signal. The ADMA does not attempt to update the CPB, as this would involve a complete master mode operation on the suspect PCI bus. The host software shall take whatever actions it can to determine the state of the bus, before attempting any accesses to the ADMA.

The PCI Interrupt interrupt signal will remain asserted until aPERR is cleared to zero by a read of ADMSTAT.

7.8 Host Operation

Figure 6 illustrates the state that Host software or BIOS may adopt when controlling ATA devices through an ADMA adapter.
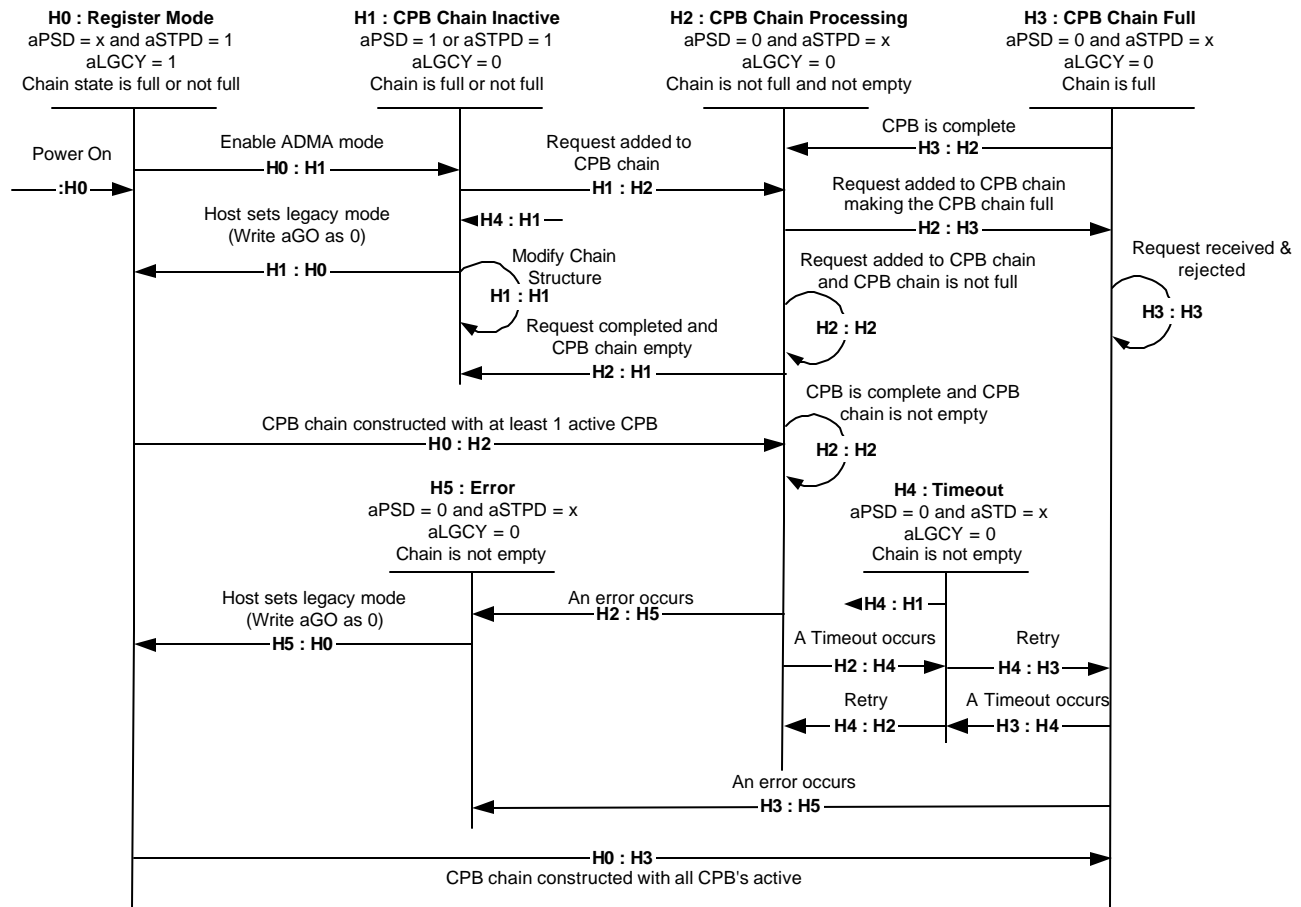
Figure 6 – Host Software States

7.8.1 Host Software States

7.8.2 H0 Register State

The host may initialize data structures and the ADMA and ATA device hardware while in this state. Error conditions are also handled in this state.

Transition H0:H1: Register to CPB Chain Inactive

The host writes a one to aGO with no valid CPBs in the CPB chain.

Transition H0:H2: Register to CPB Chain Processing

The host receives an I/O request. The host constructs CPB and PRD structures and may update the CPB lookup table if necessary. Finally aGO is written as one to inform the ADMA that a request has been added on the way to the chain.

Transition H0:H3: Register to CPB Full

The host receives an I/O request. The host constructs CPB and PRD structures and may update the CPB lookup table if necessary. The host then writes a one to aGO transitioning the host to the CPB full state. This CPB is the last one that can be added to the existing CPB structure.

7.8.3 H1: No CPB Chain Processing

The ADMA is in the paused or stopped state.

Transition H1:H1: Modify Chain Structure

The modification of the data structures such as adding or deleting CPBs may be undertaken by the host.

Transition H1:H2: CPB Chain Inactive Processing to CPB Processing

The host receives an I/O request. The host constructs CPB and PRD structures and may update the CPB lookup table if necessary. Finally aGO is written as one to inform the ADMA that a request has been added to the chain.

Transition H1:H0: Host Sets Register State

The host writes a zero to aGO transitions the host to the Register state.

7.8.4 H2: Chain Processing State

In this state the host is able to satisfy new I/O requests and add them to the CPB structure as well as process completed CPBs.

Transition H2:H3: Chain Processing to Chain Full

The host receives an I/O request. The host constructs CPB and PRD structures and may update the CPB lookup table if necessary. Finally aGO is written as one to inform the ADMA that a request has been added to the chain. This CPB is the last one that can be added to the existing CPB structure.

Transition H2:H2: Request Added

The host receives an I/O request. The host constructs CPB and PRD structures and may update the CPB lookup table if necessary. Finally aGO is written as one to inform the ADMA that a request has been added to the chain.

Transition H2:H2: CPB Completed

The assertion of PCI interrupt and no error conditions exist in the ADMA Status register causes this transition. The host examines the CPB chain to determine which CPB(s) have completed; the requester is informed of the completion(s).

Transition H2:H1: CPB Chain Processing to CPB Chain Inactive

When all requests have been completed and the CPB chain has no valid CPB's the host shall transition to the CPB Chain Inactive state

Transition H2:H4: CPB Chain Processing to Timeout

A command timeout in the CPB list causes this transition.

Transition H2:H5: Chain Processing to Error State

The assertion of PCI interrupt and aCPBERR being set to one transitions from the Processing to Error states.

7.8.5 H3: CPB Chain Full State

Transition H3:H2: CPB Completed

The assertion of PCI interrupt and no error conditions exist in the ADMA Status register causes this transition. The host examines the CPB chain to determine which CPB(s) have completed; the requester is informed of the completion(s).

Transition H3:H3: Request Rejected

The host may request the addition of an I/O request to the CPB chain. In this case the CPB chain is full and the host rejects the request.

Transition H3:H4: Timeout Occurred

A command timeout in the CPB list causes this transition.

Transition H3:H5: Error Occurs

The assertion of PCI interrupt and aCPBERR being set to one transitions from the Chain Full to Error states.

7.8.6 H4: Timeout State

This state is entered when a timer causes an interrupt. The CPB chain is examined to determine which CPB(s) has not been processed within a pre-determined time.

Transition H4:H3: Retry

The host may determine that more time is needed for a CPB to complete. The timeout timer shall be reset and the host returns to the CPB Chain Full State.

Transition H4:H2: Retry

The host may determine that more time is needed for a CPB to complete. The timeout timer shall be reset and the host returns to the CPB Chain Processing State.

Transition H4:H1: Timeout to Inactive

The host may determine that it may need to reset or perform other diagnostic activities. To do this the it has to return to the Register state. It would set aPSE to one and poll aPSD. If aPSD transitions to one the Inactive stae is entered. If aPSD is not entered within a pre-determined period the ADMA is reset and Register state is entered.

7.8.7 H5: Error State

This state is entered when PCI interrupt is asserted and aCPBERR is set to one. The host may examine the CPB chain to determine if any CPB's have completed, at least one should indicate an error condition. Any CPB's that have completed but not finalized shall have their status notified to the requestor.

Transition H5:H0: Error to Register State

The host shall write a zero to aGO transitioning to the Register state.

7.9 Resets

7.9.1 PCI Reset

PCI Reset resets the ADMA engine, and asserts the ATA RESET signal.

7.9.2 ADMA Reset (aRSTADM)

aRSTADM resets the ADMA engine but does not assert ATARESET; this is the same as power-on. Asserting aRSTADM sets ADMA engine reverts to the Legacy Idle State (see 7.7.2). Registers and data transfers are in an unknown state. The host should reinitialize all ADMA register values (not PCI register values) for this channel. If the device(s) on the channel have commands outstanding the channel should then be reset and reinitialized using aRSTA. Before resetting the ADMA engine using aRSTADM host software should attempt to stop any outstanding transfers, see A.10.

7.9.3 ATA Channel Reset (aRSTA)

Setting aRSTA to one asserts the ATA reset signal, clearing aRSTA to zero de-asserts the ATA reset signal. The interval between the two should be at least the minimum specified in the relevant ATA standard.

Annex A Programming Guidelines (Informative)

A.1 Introduction

This section is intended to review some aspects of programming in ADMA Mode that might not be immediately obvious. It makes the assumption that readers have familiarized themselves with the rest of the document.

In the past PC Compatible Legacy ATA Adapters have been accessed using specific I/O addresses (1Fxh, 17xh) and a specific set of IRQs (14, 15) where each ATA channel has a dedicated IRQ. The ADMA adapter in its ATA Register Mode uses a single interrupt for both channels. Therefore, the host should not place the I/O BARs at the Legacy I/O addresses as this may cause the host OS to confuse the ADMA adapter with an ATA Legacy adapter. This is why the ADMA uses a PCI sub-class code of 05h as opposed to 01h (see 7.4.1.6).

A.2 Programming the ADMA

The ADMA facilitates transfer of commands and data between the PCI bus and the ATA bus. This is done by a command chaining technique discussed in Section 7.2. This command chain is dependant upon the ADMA hardware, and data structures residing within host memory. It is the responsibility of the host software to allocate and initialize these structures before commencing data transfers via the ADMA.

A.2.1 PCI Configuration Header Registers

The PCI Configuration Header Registers, discussed in Section 7.4.1, are initialized by the host OS or BIOS at system boot. The ADMA Registers, discussed in Section 7.4.1.20.1, for both channels X and Y are addressed through BAR 4 in the PCI configuration header registers. The host software is not responsible for initializing the PCI Configuration Header Registers.

A.2.2 CPB Chain

The CPB chain, discussed in Section 7.6.1, may be constructed as a circular linked-list of CPB structures. A CPB chain is constructed of one or more CPBs. The CPB chain should be allocated and initialized by the host software before entering ADMA Mode.

The ADMA CPB Next Address Register should be initialized with a pointer to a CPB in the chain.

ATA register fields are used by the ADMA to write to the ATA device's Command and Control Registers. The Wait-Not-Busy (WNB) Bit set to one instructs the ADMA to read the ATA device's Status Register, and wait until the device is not busy before writing the ATA register. WNB should be set to one on the first register write for each ATA command, as the ADMA may not wait for the device to become not busy before writing the ATA register. If there are fewer than four valid entries in an ATA register field, the IGN Bit is used to indicate entries to be ignored. Note that IGN should not be set to one if END is set to one (see 7.6.1.6).

A CPB structure may contain a variable number of ATA register writes (see 7.6.1.6). This number may be fixed, or may vary dynamically from CPB to CPB. A CPB contains up to one data transfer command.

A.2.3 APRD Chains

A APRD chain, discussed in Section 7.6.1.7, is a linked-list of APRD structures. An APRD chain is constructed of one or more APRDs. Each CPB containing an ATA data transfer command or packet command points to a valid APRD chain. An ATA command that does not involve the transfer of data does not require an APRD.

APRD chains may be allocated before entering ADMA Mode or at run time. The host software is responsible for allocating and initializing one APRD chain for each CPB transferring data in the CPB chain before making the CPB valid.

It is very important to ensure that the method, direction, and length of the transfer as indicated in the APRDs are consistent with the ATA command within the CPB. If this consistency is not maintained, the system may hang.

A.2.4 CPB Lookup Table

If overlapped or queued commands are to be used, the host software is responsible for allocating and initializing a CPB Lookup Table, discussed in Section 7.6.2. The CPB Lookup Table should be allocated and initialized before entering ADMA Mode. Each entry should be a Qword, with the low-order Dword containing the physical address a CPB, and the high-order Dword cleared to zero.

The ADMA CPB Lookup Address Register should be initialized with a pointer to the CPB Lookup Table.

This table should be updated before restarting the ADMA if the host modifies CPB the chain (see A.9).

A.3 Asynchronous Operation

The ADMA operates asynchronously from the host. This means that the ADMA may process more than one command in between the host being able to service interrupts, or even as the host is servicing one. When servicing an interrupt, the host software should search the entire CPB chain to determine if more than one CPB has been completed (with or without errors).

The host should be aware that any updates it may make to shared host memory (see Figure 2), including the CPBs, APRDs, CPB Lookup Table, data buffer areas, and also the ADMA registers "CPB Next Address Register", and "CPB Lookup Table Address Register" must be undertaken such that they will not be accessed by the ADMA at the same time (see A.9).

A.4 Memory Alignment

The ADMA engine may be designed to use Quad Word alignment. Any transfer requests that the ADMA engine receives from the host should be aligned to a Quad Word boundary. This may mean that the host software has to copy the data to/from an internal Quad word aligned buffer before/after the ADMA transfer.

A.5 Register Usage

The ADMA provides a set of I/O and memory mapped registers. The I/O mapped registers provide support for ATA Register Mode operation only, and are intended for use during the initial boot process. The memory mapped registers provide a shadow of the I/O registers, as well as the registers needed to control ADMA operation. Host software writers are encouraged to use the memory mapped registers, as the I/O mapped registers may be made obsolete in future versions of the ADMA.
ATA Register Mode

The ADMA defaults to ATA Register Mode at power up, system reset, and upon detecting an error. In ATA Register Mode the ADMA is acting as an address decoder and ATA bus timing device only. The host reads and writes an ATA device just as it would on a legacy ISA bus adapter. The I/O mapped registers are initialized by the host OS or BIOS, and usually will not be the legacy 1Fxh/3Fxh values. The ADMA does not provide separate interrupts for each channel, and thus host software in ATA Register Mode should use the PCI shared interrupt architecture.

A.6 Use of aGO

Writing a one to aGO can be thought of as a "door bell". The intent is to signal to the ADMA that the host has changed something in the CPB chain, or in ADMCTL. It does not matter that the value of aGO is already one, it is the act of writing a one into aGO in ADMCTL that provides this indication to the ADMA.

A.7 Execute Single CPB

In a development environment, it is often useful to "single-step" through a CPB chain. This is achieved by setting aPSE to one and writing aGO as one for each CPB to be executed.

NOTE -   if a service interrupt is pending, the CPB that is executed may not be the next one in the chain.

A.8 Determining the Current Status of the ADMA

The host determines the status of the ADMA by examining ADMSTAT.

 shows the expected outputs from the status register dependent on the ADMA's current state. It should be noted that when the ADMA is in the ADMA Idle State, there might be outstanding Released CPBs. This means that the state of the ADMA may change without any action by the host.

A.9 Host Pausing of the ADMA Engine

The host may pause ADMA processing of an active chain, at any time, by setting aPSE to one. Upon completion of the current CPB, the ADMA sets aDONE to one, cDONE to one, transitions to the Paused State, and may assert the PCI Interrupt signal. If the ADMA is releasing a CPB it sets cREL to one and transitions to the Paused State without asserting the PCI Interrupt signal.

The host may determine when the ADMA has transitioned into the Paused State by checking aPSD set to one.

The host may determine which CPB was just completed by reading the ADMA Current CPB Address register (CCPB). The host may then modify any of the non-Released CPB entries in the chain. The host should take care to ensure that any CPBs that are in the Released State are not invalidated by changes made to the chain or to the CPB Lookup Table.

By leaving aPSE set to one, the host may execute the next valid CPB by writing a one to aGO. See A.7. The host may continue completion of all valid CPBs in the chain by clearing aPSE and writing a one to aGO.

A.10 Host Stopping or Terminating an Active CPB

The host may force the ADMA into ATA Register Mode by clearing aGO regardless of the current state of the ADMA engine. This action immediately stops the processing of the CPB chain. If a CPB is being processed, the results will be indeterminate, and the state of the ATA device will be unknown. By pausing the ADMA first, the host may make an orderly transition to ATA Register Mode. The host may then examine the CPB chain to

determine if any Released CPBs exist, and if so, take appropriate action. The host should issue an ATA channel reset to bring the device(s) on the stopped channel to a known condition.

A.11 ADMA Interrupts

The host may determine if the ADMA caused an interrupt by examining ADMSTAT. An interrupt has occurred when aDONE and/or an error bit (aUIRQ, aCPBERR, aPERR) is set to one. Reading ADMSTAT clears all error bits and aDONE to zero and de-asserts the pending interrupt.

A.12 Chain Management

The host should ensure that there is a correctly initialized CPB chain, and if overlapped/queued operations are required a correctly initialized CPB Lookup Table, before entering ADMA Mode. A valid CPB chain should consist of one or more CPB structures with the Next CPB fields pointing to the physical host memory address of the next CPB in the chain (the Next CPB field in a chain of one CPB would point to itself). cDONE should be set to one and cREL should be cleared to zero in each CPB. The host should write the address of the first CPB into the ADMA Next CPB Address register, and the start of the contiguous CPB Lookup Table into the ADMA Lookup Table Address register.

When a CPB is ready to be processed, the host should ensure that the cREL Bit is cleared and cVLD is set to one before clearing cDONE to zero. Once cDONE is cleared to zero the ADMA is in control of the CPB. The host should not modify any CPB with cDONE cleared to zero unless the ADMA is in the Legacy Idle or Paused State. If the ADMA is in the ADMA Idle State, the host should check any CPB before modification to ensure that cREL has not been set to one. Such a CPB is in the "Released" State, and should not be manipulated by the host until cDONE has been set to one by the ADMA (see 7.6.3).

While cDONE is equal to zero, the host may attempt to stop a CPB from being processed by clearing cVLD to zero. The host should then wait to ensure that cDONE has been set to one before modifying the CPB. If a CPB is being processed when cVLD is cleared to zero, the CPB will continue to be processed to completion by the ADMA, and cIGNRD will not be set (see 7.6.1.1.3).

A.13 Error Handling

If the ADMA detects that the ATA Error Bit has been set the ADMA sets the appropriate error bits in the CPB and ADMSTAT, transitions to ATA Register Mode, and asserts an interrupt.

If the ADMA detects a PCI error, it is an indication of a severe system problem. Any transfers across the PCI bus are now suspect and may result in catastrophic failure. The ADMA ceases all ATA operations, sets aPERR in ADMSTAT to one, transitions to ATA Register Mode, and asserts the PCI Interrupt signal. The ADMA does not attempt to update the CPB, as this would involve a complete master mode operation on the suspect PCI bus. The host software should take whatever actions it can to determine the state of the bus before attempting any more accesses to the ADMA.

Other errors indicate some kind of CPB inconsistency. Data Insufficiency, Excess, and CPB error (bits cPSDEF, cPSEXC, cCPBERR in the CPB Response Byte, see 7.7.4) usually mean that the CPB and APRD were not correctly constructed, or that there has been some type of data transfer error. The one exception is data excess during a Packet data transfer. In some cases the transfer size is not known or is not an exact Qword in length. In such cases the APRD data transfer length should be rounded up to the nearest Qword and pIGEX in the APRD set to one. In this way, the ADMA does not stop on error (see 7.7.4.5.).

A.14 ATAPI Data Transfers

All ATAPI APRD chains associated with a data transfer should contain at least two APRDs. The first APRD points to the packet data and should have pDIRO set to one to indicate output to the ATA device, and pORD cleared to zero to indicate use of the PIO protocol. The subsequent APRDs should indicate the direction and transfer mode for the associated data (see 7.6.1.7).

A.15 Queued Operation

For channels supporting two overlapped/queued devices the Auto-Poll Enable (aAUTEN) bit should be set to one. This causes the ADMA to alternately select each device when the ATA bus has been released, so that either device that requires service has an opportunity to assert an interrupt.

The ATA Standard suggests that nIEN be toggled during the queued protocol. This should not be done. See 7.2.6.

During queued operation, an ATA error will abort the internal command queue in the ATA device. All Released CPBs for that device will need to be reissued. The host software should search the CPB chain and clear any cREL bits it finds set to one and then write aGO as one.

Page 53

Annex B PCI Compatibility and PCI-Native Mode Bus Master Adapter Configuration (Informative)

B.1 Introduction

The ATA adapters described in section 6 require configuration information to allow proper operation. Such parameters as ATA timing values and operational options have to be set. There are a number of variations as to how these parameters are set this annex sets out one method of setting these variables. Future revisions of this standard may contain other methods.

B.2 ATA Controller PCI Configuration Registers

This section will list out the entire supported PCI configuration space registers and "Bus Master I/O" registers, with details of individual bit definitions and their respective programming being left to the Register Programming Specifics section that follows.

B.2.1 ATA Controller PCI Configuration Registers

The configuration registers used in this method of configuration uses the PCI configuration space starting at offset 40h of the PCI configuration space, see Table 25.

Table 25 – PCI Configuration Registers

Offset	Register Name/Function	Register Type
40-41	ATA TIMING (Primary)	R/W
42-43	ATA TIMING (Secondary)	R/W
44	Device 1 ATA Timing (Primary and Secondary)	R/W
45:47	Reserved	RO
48	Ultra DMA Control Register	R/W
49	Reserved	RO
4A:4B	Ultra DMA Timing Register	R/W
4C-53	Reserved	RO
54:55	IDE I/O Configuration	R/W
56-F7	Reserved	RO
F8-FB	Manufacture's ID	RO
FC-FF	Reserved	RO

B.3 ATA PIO and DMA Mode Timing and Control Registers

The following register bit layout maps define the specific ATA controller device timing and mode configuration registers. These registers control PIO timings as well as Single and Multi-Word DMA timings for device 0 on the primary and secondary controllers.

B.3.1 IDETIMx — ATA Timing Register

Address Offset: ATATIM1, Primary Channel—PCI Config. Offset 40-41h

ATATIM2, Secondary Channel—PCI Config. Offset 42-43h

Default Value: 00h

Attribute: Read/Write

This register controls timing and enable of the PIO and DMA enables/disables bus master capability for the ATA function and provides direction control for the ATA DMA transfers. This register also provides bits that software uses to indicate DMA capability of the ATA device, see Table 26.

Table 26 – ATA Timing Register

Bit	Description
15	ATA Decode Enable (R/W). This bit enables (when set to 1) or disables (when cleared to 0) decoding of the I/O addressing ranges assigned to this controller.
14	Device 1 ATA Timing Register Enable (R/W). This bit enables (when set to 1) or disables (when cleared to 0) the Device 1 ATA Timing Register.
13:12	IORDY Sample Mode (R/W). Sets the setup time before IORDY is sampled. The bit mappings are: 00: PIO-0 01: PIO-2, SW-2 10: PIO-3, PIO-4, MW-1, MW-2 11: Reserved
11:10	Reserved.
9:8	Recovery Mode (R/W). Sets the hold time after IORDY is sampled. The bit mappings are: 00: PIO-0, PIO-2, SW-2 01: PIO-3, MW-1 10: Reserved 11: PIO-4, MW-2
7	DMA Timing Enable Only Select 1. This bit enables (when set to 1) or disables (when cleared to 0) the device timings for DMA operation for device 1.
6	ATA/ATAPI Device Indicator 1. This bit indicates presence of an ATA device (when set to 1) or presence of an ATAPI device (when cleared to 0) for device 1.
5	IORDY Sample Point Enabled Select 1. This bit enables (when set to 1) or disables (when cleared to 0) the IORDY sample point capabilities for PIO transfers for device 1. IORDY is always enabled for PIO4 and PIO3, and when a PIO2 device indicates IORDY capabilities.
4	Fast Drive Timing Select 1. This bit enables (when set to 1) or disables (when cleared to 0) the Fast Drive Timing capabilities for PIO transfers, which enables faster than PIO-0 timing modes for device 1.
3	DMA Timing Enable Only Select 0. This bit enables (when set to 1) or disables (when cleared to 0) the device timings for DMA operation for device 0.
2	ATA/ATAPI Device Indicator 0. This bit indicates presence of an ATA device (when set to 1) or presence of an ATAPI device (when cleared to 0) for device 0.
1	IORDY Sample Point Enabled Select 0. This bit enables (when set to 1) or disables (when cleared to 0) the IORDY sample point capabilities for PIO transfers for device 0. IORDY is always enabled for PIO4 and PIO3, and when a PIO2 device indicates IORDY capabilities.
0	Fast Drive Timing Select 0. This bit enables (when set to 1) or disables (when cleared to 0) the Fast Drive Timing capabilities for PIO transfers, which enables faster than PIO-0 timing modes for device 0.

The IORDY Sample Point Enable Select bit is enabled (set to one) depending on the current mode, capabilities of the device, and the device's capabilities for PIO Mode 2. This bit is always enabled for IORDY Modes (PIO3 and greater). The value of this bit has no effect when Fast Timing Bank Select is Disabled.

The DMA Timing Enable Only Select bit is enabled (set to one) if and only if the device's PIO capability is much slower than it's DMA capability.

The Pre-fetch and Posting Enable Select bit is enabled (set to one) if and only if the device is a fixed disk and the device supports Mode 2 or greater.

B.3.2 SATR — Device 1 ATA Timing Register

Address Offset: Primary and Secondary Channel—PCI Config. Offset 44h

Default Value: 00h

Attribute: Read/Write

This register controls timing of the PIO and DMA capability for device 1 on the ATA function, see Table 27.

Table 27 – Device 1 ATA Timing Register

Bit	Description
6-7	Secondary Device 1 IORDY Sample Mode (R/W). Sets the setup time before IORDY is sampled. The bit mappings are: 00: PIO-0 01: PIO-2, SW-2 10: PIO-3, PIO-4, MW-1, MW-2 11: Reserved
4-5	Secondary Device 1 Recovery Mode (R/W). Sets the hold time after IORDY is sampled. The bit mappings are: 00: PIO-0, PIO-2, SW-2 01: PIO-3, MW-1 10: Reserved 11: PIO-4, MW-2
2-3	Primary Device 1 IORDY Sample Mode (R/W). Sets the setup time before IORDY is sampled. The bit mappings are: 00: PIO-0 01: PIO-2, SW-2 10: PIO-3, PIO-4, MW-1, MW-2 11: Reserved
0-7	Primary Device 1 Recovery Mode (R/W). Sets the hold time after IORDY is sampled. The bit mappings are: 00: PIO-0, PIO-2, SW-2 01: PIO-3, MW-1 10: Reserved 11: PIO-4, MW-2

The ATA Decode Enable field determines whether the cable or not is enabled:
Primary Controller: ICH Function 1 PCI Config. Offset 40h, bit 15
Secondary Controller: ICH Function 1 PCI Config. Offset 42h, bit 15
The IORDY Sample Mode and Recovery Mode fields select the current device timing cycle times.
The Device 1 ATA Timing Register bit is:

This bit is disabled by default. This bit needs to be enabled to take advantage of the independent device 1 timing register when a device 1 is attached to the cable. When this field is disabled, the Device 1 ATA Timing Register is disabled

B.4 Ultra DMA Configuration of Timing and Control Registers

The following register bit layout maps define the specific ATA controller device timing and mode configuration registers for Ultra DMA operation on all devices. These registers are programmed in systems that contain devices that implement the Ultra DMA Protocol. These registers allow Ultra DMA to be used when DMA operation is initiated by the host software.

B.4.1 UDMAC — UDMA Control Register

Address Offset: Primary and Secondary Channel—PCI Config. Offset 48h

Default Value: 00h

Attribute: Read/Write

This register controls enable for UDMA on each device. When a bit is turned on, the associated device will run in UDMA when DMA transfers are invoked, and when cleared the device will run in DMA, see Table 28.

Table 28 – UDMA Control Register

Bit	Description
4-7	Reserved
3	Secondary Drive one (Device 1) Ultra DMA Mode Enable.
2	Secondary Drive zero (Device 0) Ultra DMA Mode Enable.
1	Primary Drive one (Device 1) Ultra DMA Mode Enable.
0	Primary Drive zero (Device 0) Ultra DMA Mode Enable.

The Ultra DMA Enable bit specifies the current Ultra DMA enabled status. Ultra DMA mode is disabled by default and shall be enabled in order to take advantage of the Ultra DMA timings. When this field is disabled, the Ultra DMA Timing Register is disabled.

B.4.2 UDMATIM — UDMA Timing Register

Address Offset: Primary and Secondary Channel—PCI Config. Offset 4A-4Bh

Default Value: 00h

Attribute: Read/Write

This register controls enable for UDMA on each device. When a bit is turned on, the associated device will run in UDMA when DMA transfers are invoked, and when cleared the device will run in DMA, see Table 29.

Table 29 – UDMA Timing Register

Bit	Description
14-15	Reserved
12-13	Secondary Drive one (Device 1) Ultra DMA Cycle Time (SCT1): SCB1-66 = 0 (and) SCB1-66 = 1 (and) SCB1-66 = X (and) SCB1-100 = 0 SCB1-100 = 0 SCB1-100 = 1 (33MHz base clock) (66MHz base clock) (100MHz base clock) 00: UDMA mode 0 00: Reserved 00: Reserved 01: UDMA mode 1 01: UDMA mode 3 01: UDMA mode 5 10: UDMA mode 2 10: UDMA mode 4 10: Reserved 11: Reserved 11: Reserved 11: Reserved
10-11	Reserved
8-9	Secondary Drive zero (Device 0) Ultra DMA Cycle Time (SCT0): SCB0-66 = 0 (and) SCB0-66 = 1 (and) SCB0-66 = X (and) SCB0-100 = 0 SCB0-100 = 0 SCB0-100 = 1 (33MHz base clock) (66MHz base clock) (100MHz base clock) 00: UDMA mode 0 00: Reserved 00: Reserved 01: UDMA mode 1 01: UDMA mode 3 01: UDMA mode 5 10: UDMA mode 2 10: UDMA mode 4 10: Reserved 11: Reserved 11: Reserved 11: Reserved
6-7	Reserved
4-5	Primary Drive one (Device 1) Ultra DMA Cycle Time (PCT1): PCB1-66 = 0 (and) PCB1-66 = 1 (and) PCB1-66 = X (and) PCB1-100 = 0 PCB1-100 = 0 PCB1-100 = 1 (33MHz base clock) (66MHz base clock) (100MHz base clock) 00: UDMA mode 0 00: Reserved 00: Reserved 01: UDMA mode 1 01: UDMA mode 3 01: UDMA mode 5 10: UDMA mode 2 10: UDMA mode 4 10: Reserved 11: Reserved 11: Reserved 11: Reserved
2-3	Reserved
0-1	Primary Drive zero (Device 0) Ultra DMA Cycle Time (PCT0): PCB0-66 = 0 (and) PCB0-66 = 1 (and) PCB0-66 = X (and) PCB0-100 = 0 PCB0-100 = 0 PCB0-100 = 1 (33MHz base clock) (66MHz base clock) (100MHz base clock) 00: UDMA mode 0 00: Reserved 00: Reserved 01: UDMA mode 1 01: UDMA mode 3 01: UDMA mode 5 10: UDMA mode 2 10: UDMA mode 4 10: Reserved 11: Reserved 11: Reserved 11: Reserved

The Ultra DMA Cycle Time Field specifies the current Ultra DMA timing mode. This field only applies if the corresponding Ultra DMA Enable field is set.

B.4.3 ATAIOCFG — ATA I/O Configuration Control Register

Address Offset: Primary and Secondary Channel—PCI Config. Offset 54-55h

Default Value: 00h

Attribute: Read/Write

This register control clock selection and cable reporting for UDMA on each device. Included are tri-state control and miscellaneous PIO functionality enabling.

Table 30 – UDMA Control Register

Bit	Description
20-32	Reserved

18-19	Secondary ATA Signal Mode (SEC_SIG_MODE). These bits are used to control mode of the Secondary ATA signal pins for mobile swap bay support in mobile implementations. These bits should always be set to 00b for desktop implementations.
	    00 = Normal (Enabled)
	    01 = Tri-state (Disabled)
	    10 = Drive Low (Disabled)
	    11 = Reserved

16-17	Primary ATA Signal Mode (PRIM_SIG_MODE). These bits are used to control mode of the Primary ATA signal pins for mobile swap bay support in mobile implementations. These bits should always be set to 00b for desktop implementations.
	    00 = Normal (Enabled)
	    01 = Tri-state (Disabled)
	    10 = Drive Low (Disabled)
	    11 = Reserved

15	Secondary Drive 1, 100MHz Base Clock (SCB1-100) . Selects the 100MHz clock for UDMA on the secondary device 1 when set to 1. Selects the 66/33MHz clock for UDMA when cleared to 0.
14	Secondary Drive 0, 100MHz Base Clock (SCB0-100) . Selects the 100MHz clock for UDMA on the secondary device 0 when set to 1. Selects the 66/33MHz clock for UDMA when cleared to 0.
13	Primary Drive 1, 100MHz Base Clock (PCB1-100) . Selects the 100MHz clock for UDMA on the primary device 1 when set to 1. Selects the 66/33MHz clock for UDMA when cleared to 0.
12	Primary Drive 0, 100MHz Base Clock (PCB0-100) . Selects the 100MHz clock for UDMA on the primary device 0 when set to 1. Selects the 66/33MHz clock for UDMA when cleared to 0.
11	Reserved
10	Vendor Specific. (May be set to either a one or zero based on vendor-specific recommendation).
8-9	Reserved
7	Secondary Drive one Cable Report (SCR1). BIOS indication flag for reporting the cable type to host software. When set to 1, an 80-conductor cable is present for the device. When cleared to 0, a 40-conductor cable is present for the device.
6	Secondary Drive zero Cable Report (SCR1). BIOS indication flag for reporting the cable type to host software. When set to 1, an 80-conductor cable is present for the device. When cleared to 0, a 40-conductor cable is present for the device.
5	Primary Drive one Cable Report (SCR1). BIOS indication flag for reporting the cable type to host software. When set to 1, an 80-conductor cable is present for the device. When cleared to 0, a 40-conductor cable is present for the device.
4	Primary Drive zero Cable Report (SCR1). BIOS indication flag for reporting the cable type to host software. When set to 1, an 80-conductor cable is present for the device. When cleared to 0, a 40-conductor cable is present for the device.
3	Secondary Drive 1, 66MHz Base Clock (SCB1-66) . Selects the 66MHz clock for UDMA on the secondary device 1 when set to 1. Selects the 33MHz clock for UDMA when cleared to 0.
2	Secondary Drive 0, 66MHz Base Clock (SCB0-66) . Selects the 66MHz clock for UDMA on the secondary device 0 when set to 1. Selects the 33MHz clock for UDMA when cleared to 0.
1	Primary Drive 1, 66MHz Base Clock (PCB1-66) . Selects the 66MHz clock for UDMA on the primary device 1 when set to 1. Selects the 33MHz clock for U6DMA when cleared to 0.
0	Primary Drive 0, 66MHz Base Clock (PCB0-66) . Selects the 66MHz clock for UDMA on the primary device 0 when set to 1. Selects the 33MHz clock for UDMA when cleared to 0.

Base Clock bit(s) specify if the UDMATIM register indicates Ultra DMA 100, Ultra DMA 66 or Ultra DMA/33 timings.

Cable Reporting bit(s) specifies the presence of an 80-conductor cable (set to "1") or 40-conductor cable (cleared to "0"). This information is to be filled in by system BIOS, and interpreted by both BIOS and host software. No drive is to be programmed to Ultra DMA modes 3, 4 or 5 unless an 80-conductor cable is present. In the presence of a 40-conductor cable, all devices shall be limited to Ultra DMA mode 2 or less.